

**A8315
Combat
& A8000**

**Automatic
Test Equipment**

**SYSTEM
REFERENCE
MANUAL**



WAYNE KERR

MP MASTER



MA24504

A8315 COMBAT
& A8000
AUTOMATIC TEST EQUIPMENT

SYSTEM REFERENCE MANUAL
Including Supplement on
Automatic Digital Diagnostics (ADD)

The reference number of this publication is TP218
Including Supplement (ADD) of TP217

WKR Limited, Durban Road
Bognor Regis, West Sussex, PO22 9RL

Telephone Bognor Regis (0243) 825811
Telex 86120 WAYKER G



Digitized by the Internet Archive
in 2026

https://archive.org/details/manualsplus_12271

CONTENTS

1.0	SYSTEMS OVERVIEW	Page 1-1
2.0	THE CONTROL BOX	2-1
	Mandatory control box modules	2-1
	Optional control box modules	2-2
3.0	THE TEST CONSOLE	3-1
	The test rack	3-1
	The test jig	3-1
	Test program modules (TPM)	3-2
3.1	OPERATING INSTRUCTIONS	3-2
	The operator's console	3-2
	Installing test modules	3-6
	Installing test jigs	3-6
	Internal access	3-7
	Switching on	3-7
	Loading	3-8
	Calibrating	3-8
	Running programs	3-9
	Operator messages	3-9
4.0	SYSTEM CALIBRATION	4-1
	Selecting uncalibrated operation	4-3
5.0	TEST PROGRAM STORAGE	5-1
5.1	DISC REFERENCING	5-1
6.0	TEST JIG (TEST PLATFORM)	6-1
6.1	TEST MODULES	6-1
6.2	TEST MODULE ADDRESSING	6-3
6.3	TEST JIG WIRING	6-4
6.3.1	Ground connections for functional testing	6-4
6.3.2 (A)	Wiring for SOM continuity testing	6-5
6.3.2 (B)	ZTMs for continuity testing (A8315 only)	6-6
6.3.3	Wiring for in-circuit testing	6-6

6.3.4	Wiring for functional testing	Page 6-6
6.3.5	Wiring for combined in-circuit/functional testing using ITMs	6-7
6.3.6	Combined in-circuit/functional testing using ZTMs	6-8
6.3.7	Cross-talk	6-8
7.0	TEST PROGRAM MODULE	7-1
	EPROM Specification	7-1
8.0	TEST PROGRAM GENERATORS	8-1
8.1	FUNCTIONAL PROGRAM GENERATOR (FPG)	8-2
	Function? (List and description)	8-4
8.2	FUNCTIONAL LINE EDITOR	8-12
8.3	SOM CONTINUITY PROGRAM GENERATOR (CPG)	8-15
9.0	INTRODUCTION TO PROGRAMMING	9-1
9.0.1	Control software	9-1
	Floating-point numerical stores	9-1
	Test flags	9-1
	User flags	9-1
	Literal store	9-2
	String variable stores	9-2
	Numerical storage capacity	9-2
9.0.2	Programming	9-3
	Declarative statements	9-3
	Executable statements	9-3
9.0.3	Central measurement system	9-6
	Voltage/Time	9-6
	Input circuit	9-6
	Range selection	9-7
	Integration time	9-7
	Timing measurements	9-8
	Frequency measurements	9-9
	Other functions	9-9
9.0.4	General notes on drafting programs	9-9

9.0.5	Time out	Page 9-11
9.1	DECLARATIVE STATEMENTS	9-12
9.2	EXECUTABLE STATEMENTS	9-13
9.3	SOM CONTINUITY PROGRAMMING	9-14
9.4	INTERACTIVE PROGRAMMING	9-16
9.5	IN-CIRCUIT PROGRAMMING	9-17
9.5.1	Introduction	9-17
9.5.2	Drive levels and sense ranging	9-17
9.5.3	Resistance checks	9-19
9.5.4	Capacitance checks	9-20
9.5.5	Electrolytic capacitors	9-20
9.5.6	Inductance checks	9-20
9.5.7	Impedance checks	9-20
9.5.8	Accuracy of R,C,L & Z	9-21
9.5.9	Diode testing	9-22
9.5.10	Biodes	9-25
9.5.11	Transistor testing	9-25
9.5.12	Transistor error reporting	9-26
9.5.13	Single direction transistor testing	9-26
9.5.14	ZTM connections	9-27
9.5.15	Reduced level testing	9-27
9.5.16	Delays for active tests	9-29
9.5.17	ZTM application	9-29
9.5.18	In-circuit testing of transformers	9-32
9.6	ANALOG PROGRAMMING	9-35
9.6.1	Introduction	9-35
9.6.2	Example - simple circuit	9-36
9.6.3	<u>Continuity using ZTMs</u>	9-45
9.7	DIGITAL PROGRAMMING	9-49
9.7.1	Introduction	9-49
9.7.2	Fundamentals	9-49
9.7.3	Pin referencing	9-50
9.7.4	Basic programming	9-52
9.7.5	Initialisation	9-55
9.7.6	Program loops & algorithms	9-57

9.7.7	Mixed family logic	Page 9-58
9.7.8	Time measurements	9-58
9.7.9	Hybrid PCB testing	9-59
9.7.10	Program statement timing/operation	9-60
9.7.11	Decimal-Hex conversion	9-61
9.8	IEEE - 488 CONTROL FACILITIES	9-65
9.8.1	Syntax and specifications	9-66
9.9	TEST STATEMENT SYNTAX & SPECIFICATIONS	9-71
9.9.1	VERIFY Statements	9-72
9.9.2	VERIFY Specifications	9-73
9.9.3	MEASURE Statements	9-74
9.9.4	MEASURE Specifications	9-75
9.9.5	ADJUST Statements	9-76
9.9.6	ADJUST Specifications	9-77
9.9.7	RELAY Statements	9-78
9.9.8	RELAY Specifications	9-78
9.9.9	Truth table of auto app'd flag settings	9-79
9.9.10	STIMULUS/POWER SUPPLY Statements	9-79
9.9.11	STIMULUS/POWER SUPPLY Specifications	9-80
9.9.12	CONTROL Statements	9-83
9.9.13	CONTROL Specifications	9-84
9.9.14	TIMER Statements	9-87
9.9.15	TIMER Specifications	9-88
9.9.16	ZTM Statements	9-91
9.9.17	ZTM Specifications	9-92
9.9.18	DIGITAL Statements	9-93
9.9.19	DIGITAL Specifications	9-94
9.9.20	String Store Statements	9-95
9.9.21	String Store Specifications	9-95
9.9.22	Using the Clock	9-99
9.9.23	Syntax Summary	9-100
9.9.24	Test Times	9-101
10.0	MESSAGES & ERRORS	10-1
10.1	OPERATOR MESSAGES & APPLICATION ERRORS	10-1
10.2	SYSTEM ERRORS	10-19

11.0	USER INFORMATION	Page 11-1
11.1	Routine checks	11-1
11.2	Floppy disc	11-2
11.3	A8315 Debugger	11-3
11.4	Test Jig Accessories	11-8
11.5	VDUs in use with the system	11-9
11.6	Transit packaging	11-9
12.0	SUPPORT & SERVICES	12-1
Appendix A	TEST MODULE SPECIFICATIONS	A-1
A.1	Analog test module (ATM)	A-1
A.2	Power supply module (PSM)	A-4
A.3	Voltage/current source module (VIM)	A-6
A.4	Sinewave source module (SSM)	A-8
A.5	Reed relay module (RRM)	A-9
A.6	Timer interface module (TIM)	A-10
A.7	Tristate impedance test module (ZTM)	A-11
A.8	Pulse generator module (PGM)	A-14
A.9	Shorts & opens module (SOM)	A-16
A.10	Logic test module (LTM)	A-17
A.11	Digital test module (DTM)	A-19
A.12	Clock/calendar module (CCM)	A-21
A.13.1	Program link module (PLM)	A-22
A.13.2	Prom enable module (PEM)	A-22
A.14	High speed extended maths	A-23
A.15	Mini-disc	A-24
A.16	SIO options	A-25
A.17	IEEE control interface	A-26
A.18	Automatic digital diagnostics (ADD)	A-27
A.19	Footswitch option (FSI)	A-29
A.20	Customer interface module (CIM)	A-30
A.21	Vacuum adaptor for test platform	A-31
A.22	Real Time test option	A-32
A.23	Data compression module (DCM)	A-33
A.24	Clock qualifier module (CQM)	A-34

Appendix B1	INSTALLATION	Page B-1
Appendix B2	SIO MODULES - Setting-up	B-3
B2.1	Interface to ATE System	B-3
B2.2	Interface to RS232	B-3
	S203 settings	B-3
	COMP/PERI switches	B-5
	Connector wiring	B-5
B2.3	Port 2 (Aux SIO) - Specific features	B-5
	End-of-line character	B-5
	Program transfer	B-5
	Hardware Handshake	B-6
	Connector wiring	B-6
Appendix C	ABBREVIATIONS GLOSSARY	C-1
Appendix D	DIRECT POLLING OF VDU	D-1
Supplement (TP217)	AUTOMATIC DIGITAL DIAGNOSTICS (ADD)	1

ILLUSTRATIONS

Figure	Title	Page
1.1	System Block Diagram	1-3
3.1	Test Interface Connector (TIC)	3-1
3.2	Test Units and Control Box	3-3
3.3	Ticket Print-out	3-8
4.1	Dual-ramp integrator measurement	4-2
6.1	TIC - Binary Address Coding	6-3
8.1	Transmit/Receive Software Protocol	8-3
9.1	Shunt Elements	9-17
9.2	Lead Impedances	9-21
9.3	Analog Circuit Example	9-36
9.4	Flow Chart Example	9-39
9.5	Analog Cct + Pin Connections	9-40
9.6	Interface Connectors (Digital)	9-49
9.7	Digital Circuit Example	9-52
9.8	Reed Relay Connections (TIC)	9-78
D.1	SIO Card Layout	D-2
D.2	SIO Circuit Diagram	D-3

1.0 SYSTEMS OVERVIEW

The A8000 and A8315 automatic test equipments are based on a completely modular approach to provide highly flexible systems, in which the users' requirements dictate their particular configuration. The system hardware and software have been developed in parallel, with close co-operation between hardware and software engineers, to ensure that both equipments provide the user with truly versatile systems. They are easy to program, and can be used in the test mode by all grades of test and production personnel, engineers or quality assurance staff. In order that the user shall have a high degree of confidence in the integrity of the results obtained, the systems perform a series of self calibration and self checks at switch-on, and subsequently whenever it is required by the user.

The modular design of the equipments is such that users have only to include in the system those functions that are necessary to meet individual test requirements. The basis of this modularity is the bus-structure employed, which can be seen in the system block diagram (Fig 1.1).

The equipments can be configured to perform the following different types of testing:

- a) Continuity testing on both bare and assembled printed circuit boards, wiring looms, harnesses etc.
- b) In-circuit testing to ensure correct value, location, and orientation of discrete analog components.
- c) Functional testing of analog circuits.
- d) Functional testing of digital circuits.
- e) Functional testing of digital and analog (hybrid) circuits.
- f) System/sub-system testing.
- g) Combinations of continuity/in-circuit/functional testing (A8315)

A8000 employs one test jig for continuity testing and another for all in-circuit and functional tests. For this reason, each test jig carries a Test Program Module (TPM) on which the appropriate test program can be stored in EPROM. Continuity programs are generated using a Shorts and Opens Module (SOM). As an alternative to EPROM, A8000 programs can be stored on disc, with continuity tests on a separate program from all other tests.

A8315 employs tri-state in-circuit test modules (ZTMs) which permit continuity and functional tests to be combined within a single program. For this reason, only one test jig is necessary and this does not carry EPROMS, program storage always being on disc. The A8315 can also be used to generate and run SOM-based continuity programs, but these cannot be stored on EPROM.

The equipments can be used at every stage of production from bare boards to complete systems. This flexibility is achieved by sophisticated system software directly controlling the test hardware. The test program is created either by the conversational Interactive Program Generator (IPG) or by directly entering the abbreviated test statements.

As a test station, the system comprises two main units: the CONTROL BOX and the TEST CONSOLE. Full descriptions of these follow in their relevant sections.

As programming stations, the systems also require a VDU, or equivalent RS 232 compatible terminal. A programming station can also be used for testing.

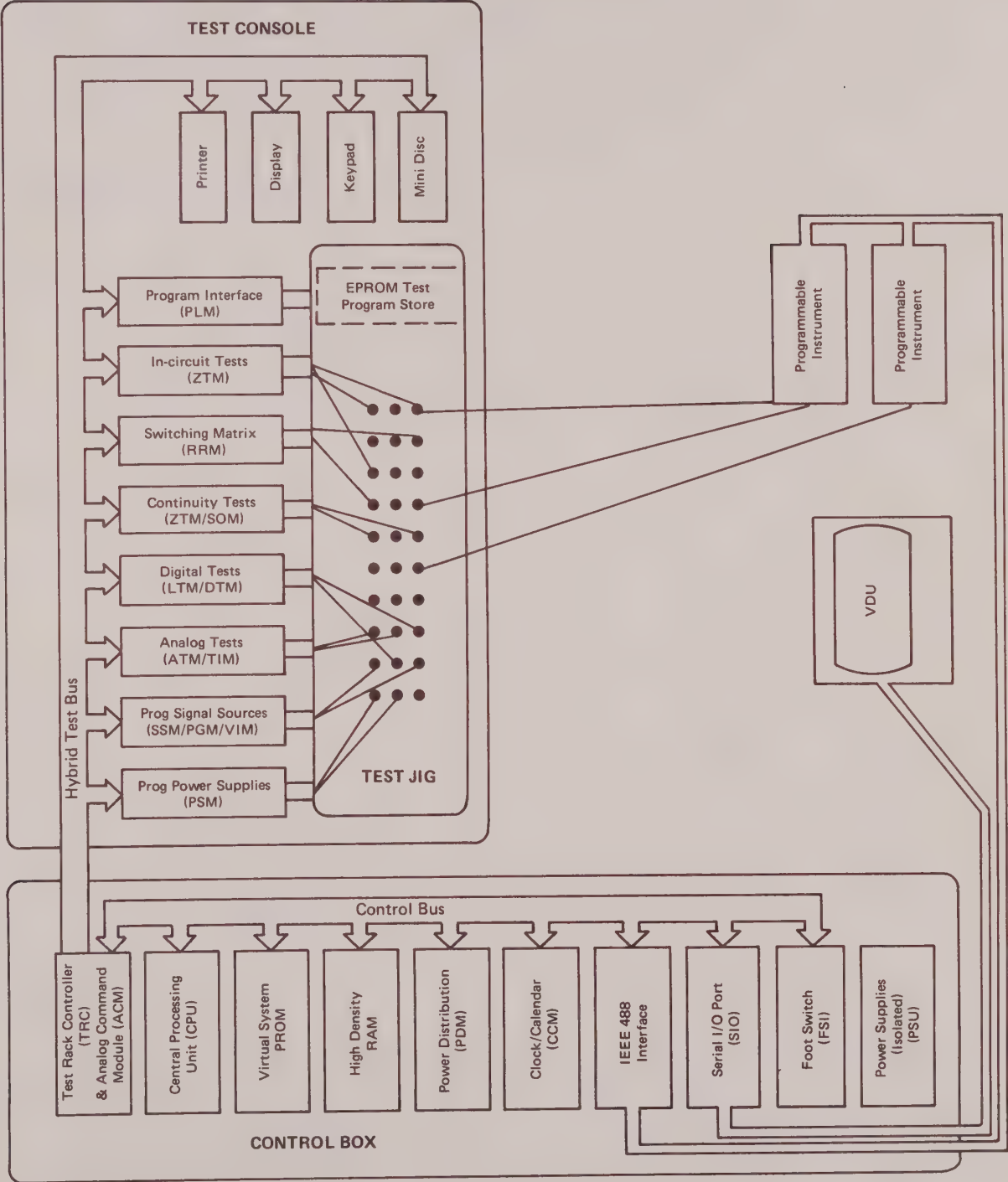


Fig. 1.1 System Block Diagram

2.0 THE CONTROL BOX

The Control Box contains the heart of the system, a powerful Z80 microprocessor which forms the core of the central processing unit (CPU). This communicates with the rest of the circuits in the control box via the Control Bus, and with the interface modules in the test console via the Hybrid Test Bus. The control box is composed of a Eurocard rack with associated sub-assemblies.

MANDATORY CONTROL BOX MODULES

Central Processing Unit (CPU)

The CPU board houses the Z80 microprocessor and associated circuits. Optionally on A8000, and as a standard feature on A8315, the board also incorporates a hardware maths chip to extend the number of mathematical functions available to the user, and to increase testing speed.

Test Rack Controller (TRC)

The Test Rack Controller can be separated into two distinct functions:

- 1) The Real Time Module (RTM) - which provides the timing functions required for accurate measurement.
- 2) The Test Bus Interface (TBI) - which provides the interface between the Hybrid Test Bus and the Control Bus.

Analog Command Module (ACM)

This module provides the main analog measurement capability of the system. By interacting with the software, with the Real Time Module and the various Test Modules, it enables the system to perform accurate analog measurements.

Virtual System PROM

This contains the system software, housed in a number of Block Switch Memory (BSM) modules. A system of memory paging is employed, enabling the size of the software to exceed the 64K directly addressable. Part of the system software is modular in an analogous manner to the modularity of the interface hardware in the test console.

High Density Ram (HDR)

This is a high density dynamic random access memory having a capacity of 32K bytes, of which half (16K) is available for program generation, editing and debugging.

Power Distribution Module (PDM)

Serves two functions:

- 1) Distributes the system power supplies to the control and hybrid test buses, and provides visual monitoring of the supplies via LEDs on its front panel.
- 2) Contains the necessary switching to enable the ACM to be calibrated automatically by the system during operation.

System Power Supplies

The system power supplies are housed in the control box, with the exception of the +30V test-bus supply, which is housed in the test console as part of the Console Power Supply (CPS).

OPTIONAL CONTROL BOX MODULES

Serial Interface (SIO)

This is an RS232 serial I/O port to enable the user to connect a VDU, Printer or any compatible I/O device. Up to two separate ports can be accessed by the user. In the case of a programming station, one of these is mandatory.

IEEE 488 Interface

This interface is used to extend the capabilities of the system to include the use of IEEE 488 compatible instruments. In this case the system acts as an IEEE controller.

User Programmable Power Supplies

Up to two dual user programmable power supplies (PSU's) may be housed in the control box. Each dual PSU requires a Power Supply Module (PSM) to be sited in the test console in the appropriate position. A third dual programmable unit can also be supplied as an external addition.

Calendar Clock Module (CCM)

This module allows such details as current time, date, program run period, etc, to be incorporated in the test sequence and to appear on the VDU, the 20-character display and on any print-out. The information is derived from an accurate crystal oscillator and the module will automatically up-date calendar information correctly, including leap-years. The only user operation should be to allocate correct local time, as the module is designed to operate correctly for extended periods even with the ATE switched off.

Foot Switch Interface (FSI)

Intended to augment the Automatic Digital Diagnostic (ADD) option, this module enables the operator to respond to the YES/NO or GO/NOGO messages related to ADD learn and diagnose routines without being involved with VDU keyboard entries.

3.0 THE TEST CONSOLE

The Test Console serves as the interface between the system, the operator, and the Unit-Under-Test (UUT). It comprises three main assemblies:

- 1) Test Rack
- 2) Test Jig
- 3) Operator's Console (Control Panel).

THE TEST RACK

This houses the optional interface modules, known as TEST MODULES. The user defines his configuration by installing test modules in the test rack.

However, to minimise the possible number of test rack population changes, the system is arranged to allow the user to operate a sub-set of the total test module population.

To prevent abuse of the system and UUT, the test program contains a description of the required test rack configuration, and this is checked against the actual configuration. However, a test module residing in the test rack is not 'visible' to the system until it is plugged forward into a jig, mating with a connector which has the required linking (See Section 6.2). Test modules may be plugged into a jig, or retained away from the jig, allowing the user to operate sub-sets of a particular test rack population.

THE TEST JIG

This is the interface between the test modules in the test rack and the UUT. The connector that provides this interface is a common connector to all modules, and is known as the Test Interface Connector (TIC). It is a 48 way DIN 41612 high quality two part connector, its pin layout being shown in Fig 3.1. The pins are referred to by their bank letter (A, B or C) and the pin position in the bank (Pins 1 to 16 inclusive). Hence pin A4 would refer to pin no 4 in Bank A.

In order to minimise the degradation of measurement quality, the test jig and the test module connection system have been designed such that the UUT and the test modules are physically close. In addition, the calibration functions within the system

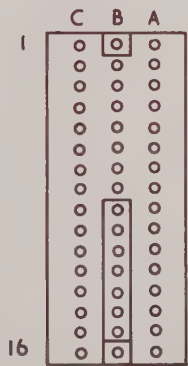


Fig. 3.1 -
Test Interface
Connector (TIC)

ensure that it is calibrated up to the point of contact with the UUT, thus providing true test site measurements for analog functions.

TEST PROGRAM MODULE (TPM)

Test program storage is described in section 5, but an important difference in this respect, between A8000 and A8315, must be mentioned here. An alternative to disc storage is available on A8000, whereby programs can be stored on EPROMs, located on the Test Program Module (TPM) fitted inside the test jig. Once the program interface (rightmost in the test rack) has been connected to a program-bearing TPM, the program can be registered (automatically on switch-on, or using LOAD/CAL button) and operated. The program interface will consist of either a PLM alone (A8315), or a PEM/PLM combination (A8000).

A8000 can be used to generate a test program, and store it either on disc or on EPROM, or both if required, and can use either storage medium as a base for operating a test program.

A8315 stores all programs on disc, but can use either disc or EPROMs as the test program operating base. The EPROMs will always be located on TPMs which will usually be mounted inside the test jig, but must always be accessed via the rightmost test rack module.

3.1 OPERATING INSTRUCTIONS

THE OPERATOR'S CONSOLE (see Fig. 3.2)

This comprises an illuminated display, a 'ticket' printer and a 'keypad'.

a) DISPLAY

The 20 character alphanumeric display is used to output messages and provide graphical position indication during the execution of an ADJUST instruction.

b) PRINTER

The electrosensitive printer is programmable to enable the user to select 5, 10, 20 or 40 characters per line. It is used for failure logging, special program messages and test program print out.

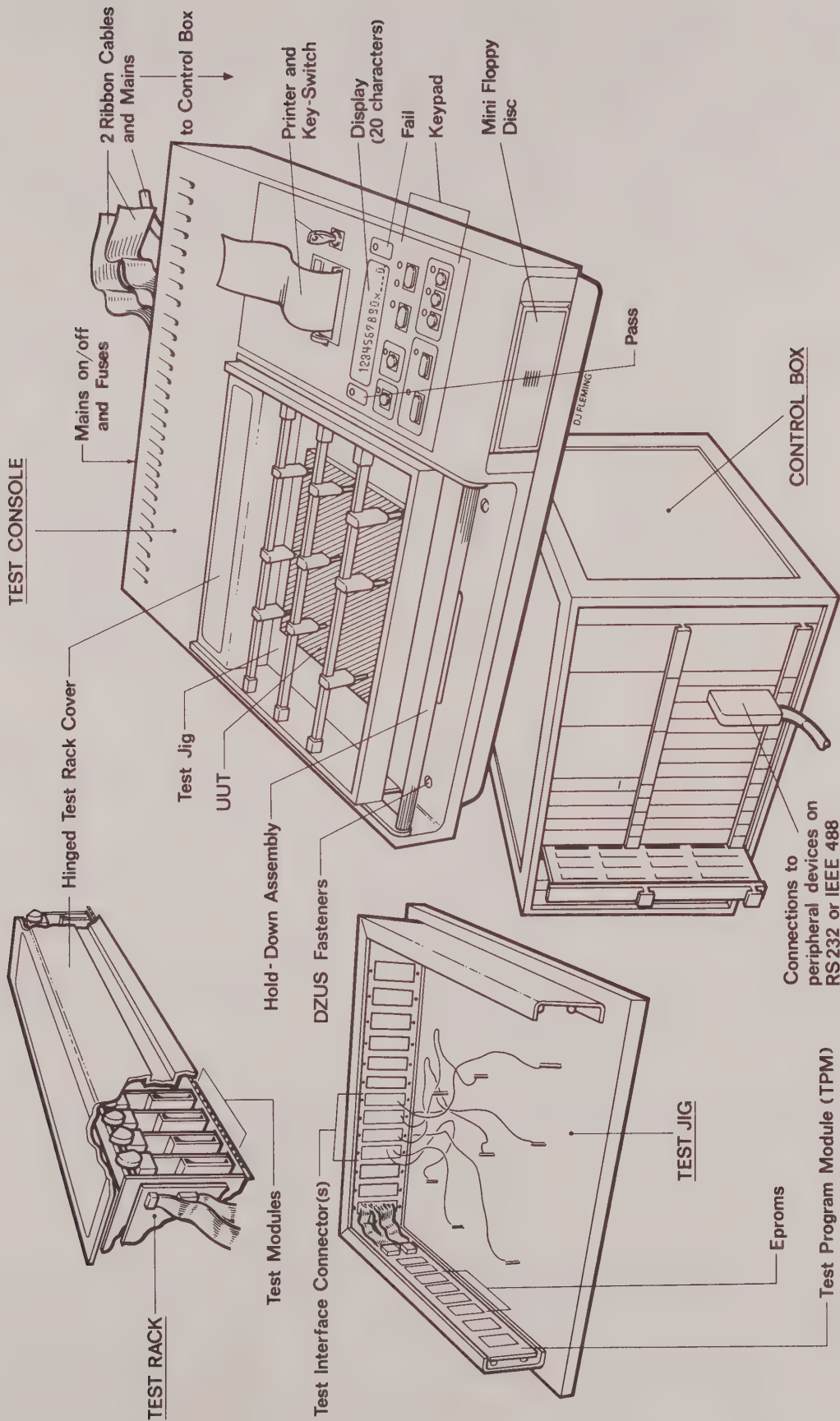


Fig. 3.2 Test Units and Control Box

c) KEY SWITCHES

The operator's keypad is equipped with nine key switches and associated LED indicators to enable the operator to control the system operation. Four of the switches operate in an alternate manner. A depression of the switch causes the mode to be invoked if not currently valid, or to be cancelled if it is valid. The mode state is indicated by the associated LED. Whether operation of a key switch causes the described action will depend on the current system state, i.e. not all operator requests are always permissible.

Power: A latching switch, used to turn the system on and off. This switch DOES NOT ISOLATE MAINS POWER FROM THE SYSTEM, but controls the distribution of mains power to the system power supplies. Mains power can be isolated by the switch on the rear connector panel of the test console. The red LED indicates that mains power is on the system, but that it is switched off. When the system is switched on, the red LED is extinguished and the yellow one illuminated.

Load/Cal: Loads the test program from the disc or TPM and checks that the correct modules for that particular program are connected to the test jig in the correct location, and performs any system calibration that may be necessary. This action should be initiated whenever the test jig is changed. The indicator is on while this is in process. Load/Cal is performed automatically at switch-on.

Proceed: Used to initiate running of a test program and to continue after an interruption. Also indicates to the system that an operator has performed a requested action.
The indicator is on whenever the system is operating and requires no operator action.

- Reset:** Used to abort the execution of a test program before the natural end of the test sequence, or to otherwise set the system into the reset ('READY') state. If the reset is requested during the execution of a program whilst in the programming mode, control will be returned to the VDU.
The indicator is on whenever the system is in its reset state.
- Program:** Switches the system into the user programming mode to accept communication via the VDU connected to the serial I/O port.
- Note An LED associated with each of the four following key switches (Single/Recycle/Stop on fail/Print all) is illuminated when the selected function is effective. A short time may elapse between operation of the switch and the availability of the function : the LED illuminates only when the selection is effective.
- Single:** Each time the PROCEED button is pressed, the system executes a decade of test statements and then stops awaiting a further PROCEED, i.e. 1-9: 10-19: etc. On A8315, Single inhibits Fast mode. On leaving Single, FAST is not automatically reinstated.
- Recycle:** The program is repeated after the completion of each test program sequence.
- Stop on fail:** The system stops the test sequence whenever a failure is detected. Pressing the PROCEED button will cause the system to resume the test sequence.
- Print all:** The system prints the number of every statement executed, together with results of any measurements or comparisons. Usually, the printer outputs

a test record, comprising test program name, program result (PASS/FAIL) and, if relevant, fault signatures, adjustment results and user requested messages.

When programming, Print All causes the full VDU dialogue to be printed. On A8315, Print All inhibits Fast mode. On leaving Print All, FAST is not reinstated automatically.

d) PASS/FAIL lamps

The panel is provided with indicator lamps which reflect the status of the current or previous execution of the test program.

e) PRINTER ON/OFF

This is a KEY operated switch to ensure that the printer cannot be switched on or off inadvertently.

INSTALLING TEST MODULES

First rotate the appropriate catch on the front of the test rack to allow insertion of the module. Ensure the module handle is in the folded position, and slide the module firmly home, rotating the catch to retain the module in that position. Removal is the reverse, using the module handle pulled out.

WARNING

Test modules must not be installed or removed with the system power on. Such action could cause damage and invalidates the system warranty. (This switch-off is not necessary when modules already installed are moved forward or backward to engage/disengage the appropriate Test Interface Connectors on the Test Jig, but this latter operation should not be performed during execution of a test program).

INSTALLING TEST JIGS

The mechanical hold-down assembly and hinged test rack cover should both be in the 'up' position and THE TEST MODULES IN THE TEST RACK MUST BE RETAINED BACK, AWAY FROM THE JIG. The test jig drops into the central well of the

test console and, once it is seated, the test modules can be released from their catches. Using the unfolding handles, pull the required test modules forward into the jig. The rear connections of the modules are spring loaded to provide assistance in this process. Which test modules are required will depend upon the test jig and UUT, but the rightmost position will be occupied by a mandatory assembly which, when connected to the test jig, will allow test jig defined disc-stored test-program access for both A8315 and A8000, EPROM-stored test-program access for both, and EPROM-based test program write facility for A8000. The folding handles should be refolded onto the modules to allow the hinged cover plate to be closed.

Removal of a jig is the reverse of the above procedure, but again THE TEST MODULES MUST BE RETAINED AWAY FROM THE JIG before jig removal.

Test jigs may be removed with power on the system. Test modules must not be moved during program execution, but only when the system is in the reset state.

NOTE : After changing the test jig, or set of modules connected to the jig, Load/Cal must be initiated.

INTERNAL ACCESS

To gain access to the inside of the test console for paper replacement or other user maintenance, remove the printer key, hold the front and rotate the two Dzus fasteners counter-clockwise. The top is hinged at the rear. Raise it as far as possible, when a self-engaging latch should be heard to operate, locking the top in the raised position. To close, hold the top firmly and lift slightly whilst releasing the catch on the supporting strut. When the Dzus fasteners have been re-located, replace the printer key.

ALWAYS ensure that no tools or other loose items remain in the Console before closing the top.

To replace the paper roll, unscrew the roll spindle and load new roll. Insert end into feed slot and rotate thumbwheel.

The system may be operated open, to allow access to the underneath of the jig during program debugging.

SWITCHING ON

BEFORE switching on, see IMPORTANT note in Appendix B1.

Press the power switch. The red LED will extinguish and the yellow one will illuminate.

LOADING

Test program registration will now take place, indicated by the message 'LOADING'. If the jig is wired for test program from disc, the system will fetch the program from the disc. If that program is not present on the disc currently in the drive unit, the directory of the disc will be printed on the ticket printer.

The population of test modules plugged into the test jig is checked against the test program requirements. If they do not agree, the message 'TEST RACK MISMATCH' will appear on the display, and the printer, if on, will output further information concerning the nature of the discrepancy, an example of which is shown in Fig. 3.3.

CALIBRATING

If no mismatch is present, the system proceeds to 'CALIBRATING'. The display may request 'INSERT CAL PLATE', in which case the operator should install the appropriate calibration connections for the jig, and then press PROCEED. The system will continue calibrating and, when complete, will display 'REMOVE CAL PLATE'. Subsequent pressing of PROCEED will enter the reset state and display 'READY'

PROGRAM IS		
424B02		
TEST RACK MISMATCH		
PRESENT IN RACK		
MOD	ADDR	
RRM	5	
ATM	6	
PSM	1	
REQUIRED BY PROGRAM		
MOD	ADDR	SLOT
RRM	5	11
SSM	4	10
ATM	7	9
PSM	1	1

Figure 3.3
Ticket Print-out

ATM calibration can be aborted by pressing RESET when 'INSERT CAL PLATE' is displayed. The system will then operate at the basic - uncalibrated - accuracy and will not invoke ERR PIN x/yy messages.

If no test program is available, the LOADING process is skipped, and the system will calibrate the test modules connected to the jig as described in the first paragraph of CALIBRATING (above).

NOTE: The reset state of the system is indicated by the reset lamp being illuminated and, in general, the message 'READY' being displayed. However, in certain cases where the reset state is entered as a

natural end to a sequence (e.g. end of a test program) rather than operation of the RESET key, the display will indicate the last relevant message (e.g. TPXXXXXX PASS), to allow operator observation 'at leisure'. 'READY' also is not displayed in the reset state when the system is in the programming mode.

RUNNING PROGRAMS

Install a UUT on the test jig and, if necessary, lower the hold-down assembly, holding the round bar. Before final engagement, ensure that the plungers are positioned so as to avoid delicate components. Engage the assembly by pushing down on the round bar until it is flush with the test console. The pressure required should not exceed 2kg.

Pressing PROCEED will initiate execution of the test program, which will proceed automatically unless operator actions are required.

At the end of the program, the display will show the program name and the result (e.g. TPXXXXXX FAIL). The Pass/Fail lamps will indicate appropriately, and the printer will output the state as the last entry of the test record print-out.

If no program 'LOADING' has been performed, pressing PROCEED will cause the message 'NO PROGRAM' to be displayed.

OPERATOR MESSAGES

A list of the most common messages appearing on the 20-character display, and their meaning, together with any required operator action, follows. After requesting an operator action, the system implicitly expects the operator to press PROCEED when the action has been performed (but see ADJUST).

<u>MESSAGE</u>	<u>MEANING</u>	<u>REQUIRED ACTION</u>
LOADING	Test program registration	None
WRONG PROG STRUCTURE	Program not valid (e.g. 2 Eproms wrong way round)	If Eprom, check physical positioning. If disc, treat as 'PERM ERROR'
CALIBRATING	Calibration	None

3-10	(section 3.1 continued)	
<u>MESSAGE</u>	<u>MEANING</u>	<u>REQUIRED ACTION</u>
INSERT CAL PLATE		Install calibration connections or press RESET to abort ATM pin calibration.
REQUESTED ATM UN-CAL	Warning message that operator has chosen to abort pin calibration	Press PROCEED to obtain 'READY' state.
REMOVE CAL PLATE		Remove calibration connections
NO PCM - PARTIAL CAL	ATM pins not calibrated, due to lack of PCM <u>or</u> PSM	Correct if required
READY	System reset	Operator choice
NO PROGRAM or PROG NAME MISMATCH	Test Program has not been registered	Press LOAD/CAL if appropriate
TEST RACK MISMATCH	Incorrect modules for the program	Plug correct modules into jig
nnnn ADJUST xxxxxx		Identify component, press PROCEED to obtain graphical display
X-----+--0----X	Graphical display	Adjust previously specified component, such that the 0 is over the +; whenever the 0 is over the + or X, the displayed character becomes a *. Providing the 0 lies between X's the ADJUST will pass.

<u>MESSAGE</u>	<u>MEANING</u>	<u>REQUIRED ACTION</u>
OVERRANGE ERROR	Test unit not providing correct signals	None
NO PULSES - TIME OUT	Test unit not providing correct signals	If program terminates with further error, report to program writer
TP cccccc PASS	Program passed	
TP cccccc FAIL!	Program failed	
nnnn MV=xxxxx	At statement nnnn, Measured Value or compared store equals xxxxx	
nnnn ERR PIN x/yy	Uncalibrated pin. ATM pin not calibrated during normal ATM calibration (see section 4.0, paragraph preceding 'SELECTING UNCALIBRATED OPERATION').	Continue if allowable by pressing PROCEED
RUNNING FAST MODE	Program instructs system to run in FAST MODE	None
<u>DISC MESSAGES</u>		
DRIVE NOT READY	No diskette in Drive Unit	Insert correct diskette in Drive Unit
xxxxxx NOT FOUND	Test program xxxxxx not present on diskette in Drive Unit	Insert correct diskette in Drive Unit

MESSAGE

MEANING

REQUIRED ACTION

PERM ERROR	}	Unrecoverable disc error has occurred	Remove diskette - re-insert and re-try - if error persists, diskette should be re-copied from master (see also section 11.2)
SECTOR NOT FOUND			
SEEK FAIL			

OTHER MESSAGES

For a full list of all messages, see Section 10.

4.0 SYSTEM CALIBRATION

The ATE uses a sophisticated software technique to calibrate the system for the user. This enables the system to maintain its high standard of accuracy through changes in ambient and specific component temperatures, effects due to component ageing, and to calibrate the measurement system up to the point of contact with the UUT.

Calibration is initiated in two ways:

- a) at switch-on automatically
- b) under operator control from:
 - (i) the operator's keypad
 - (ii) the VDU keyboard when in programming mode.

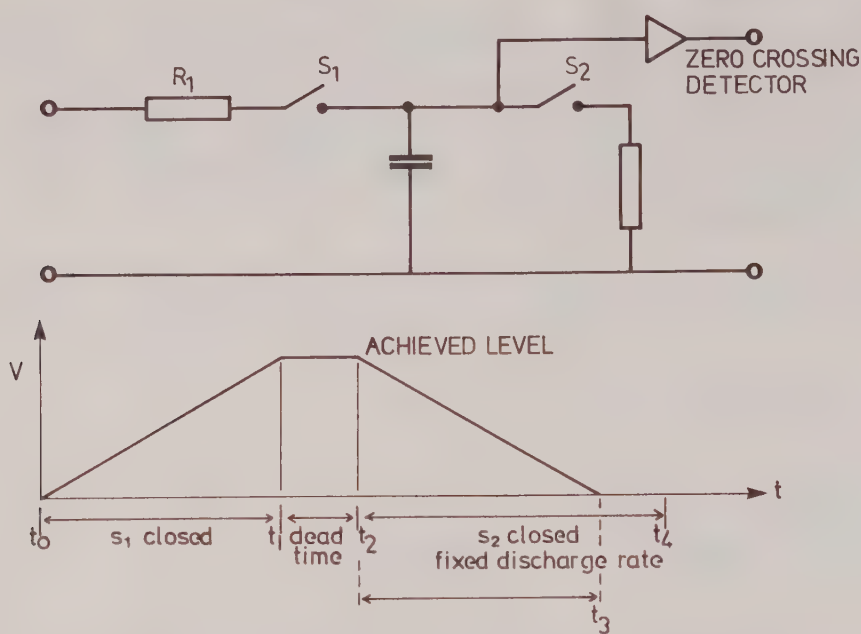
Initially test program 'loading' is performed, in which the test program is verified for structure. The program contains information regarding the required test rack configuration, and this is compared with the actual configuration. Assuming this check is successful, the system proceeds to calibrate.

If the program interface is not plugged into a jig with a valid program disc reference or EPROM-based program, the 'loading' process is not performed, and the system proceeds immediately with calibration.

Calibration checks the ability of the system to measure accurately both time-related and analog parameters. The basic analog measurement system relies on a dual-ramp integration technique, whereby an input gate is opened for a period determined by the voltage range and the integration period, such that the voltage achieved on the storage point lies ideally between 2.5 and 5.0 volts. The input integration period is determined accurately by the system and, when terminated, the system discharges the storage at a fixed rate, measuring the time taken until the stored charge reduces to a predetermined point. Thus analog measurements are made by relating their value to the accurate internal clock. The diagram (next page) illustrates the method used.

This method is used during calibration of ACM and ATM by connecting amplifier inputs to 0V, measuring the output level and retaining it as an OFFSET; inputs

are then connected to VREF, and GAIN data retained for all input combinations.



Slope determined by value of input level and value of R_1 (range setting). Achieved level determined by both above, and by $t_0 \rightarrow t_1$ period.

Measured discharge period is $t_2 \rightarrow t_3$. Rate is fixed, so measured time is related only to achieved level.

Fig. 4.1
Dual-ramp
Integrator
Measurement

Calibration can be considered as consisting of two processes. (All references to PSM also imply PCM).

1) - The first calibration will take place whatever the contents of the test rack might be. This is the calibration of the Analog Command Module, which consists of creating a set of tables which contain all the information regarding gain and offset factors within the ACM.

2) - The second calibration will take place only if the system detects the presence of an addressed Analog Test Module in the Test Rack. This second calibration also requires the presence of an addressed Power Supply Module (PSM) in the test rack to enable the internal accurate voltage reference (VREF) to be brought out to the test jig. The system will request the user to insert a calibration plate ('INSERT CAL PLATE'). This is a conducting plate that is used to direct VREF to the ATM pins. Physically it may take the form of a plate for a 'bed-of-nails' jig, but may take other physical forms depending on the physical connections to the UUT. Electrically, the calibration is achieved by the application of VREF, from pin B1 of the highest addressed PSM, to all ATM pins that will subsequently be used in the test program. This B1 pin is switched so as to appear open circuit except during calibration. It is essential to realise that a few connections must NOT

be allowed to contact the conducting plate (or other VREF connection system): these exceptions are described in the succeeding WARNING paragraph. The ATM pins may be connected to other high impedance test module pins, but this is not relevant to the Calibration. Gain factors are thus captured for all connected ATM pins, and these factors, together with the ACM information, are used to correct subsequent analog measurements. On completion of ATM calibration, the message REMOVE CAL PLATE will be displayed.

If no PSM is present the system will detect this, and indicate it to the user through the message 'NO PCM - PARTIAL CAL'. This results in the system having the basic uncalibrated accuracy. (See Appendix A for specification).

In addition, the ACM is recalibrated before each execution of the test program, to eliminate errors due to gain drift within the ACM. ATM/ACM offsets are eliminated by making an offset measurement immediately preceding the required measurement in a test program. ATM gain drift is not statistically relevant, within the specified operating temperature of the system.

NOTE- If the system is switched on from a cold temperature state, it is advisable to wait until operating temperature is reached. If this is inconvenient, the system may be used but the user is advised to recalibrate at appropriate intervals.

When PSM's and ATM's are in the test rack it is assumed that all the ATM pins used in the program will have been calibrated as outlined above. The software will detect when the program uses a pin which has not been so calibrated and displays a message on the console indicating the module and pin ('ERR PIN X/YY'). Pressing PROCEED causes the program to continue but with the basic uncalibrated accuracy for that pin.

SELECTING UNCALIBRATED OPERATION

ATM calibration can be aborted by pressing RESET when the message 'INSERT CAL PLATE' is displayed. In this condition, ATMs are not calibrated and the system operates with basic uncalibrated accuracy. This is useful for short programs or program debugging, or in circumstances where the full measurement accuracy is not required. It does not affect the calibration of the ACM, nor does it prevent the operator from requesting a full calibration at a later time. The message output is "REQUESTED ATM UN-CAL". PROCEED must be pressed to acknowledge the situation.

The operator has no control over ACM calibration, which is made automatically at switch-on, and upon request, but the programmer can - on the A8315 - elect to omit the ACM calibration normally performed when each program run is started. The effect of this is to retain the gain and offset factors obtained at the last ACM calibration (probably on switch-on) and use this data during the current program run. The pre-run ACM calibration contributes approximately two seconds to run time and, in time-sensitive areas, this time saved by omitting ACM calibration can be valuable.

To omit the pre-run ACM calibration, the programmer must use 'MODE, UNCAL' as the first program statement. For example:

```

S1    MODE, UNCAL
D10   PSM=1
D20   ATM=7
D30
.
.
.
.
S70   SU PSU PS....

```

WARNING

As already stated, some pins must be isolated from the VREF voltage (i.e. from the calibration plate or other system). These include ALL ITM pins, the +ve and -ve supplies from PCM(s), the current monitor outputs of VIM(s), the square-wave output from SMM(s) and all system ground pins.

Failure to isolate these connections will produce an erroneously-calibrated system and could cause damage. With a conducting plate, used for calibration with a bed-of-nails arrangement, holes can be drilled to provide the necessary isolation. PSM and ZTM pins, when in an initialised state, present a high impedance to VREF and may be safely connected to the conducting plate. In the case of edge connectors, relays may be used to provide the disconnections during calibration. It is good testing practice to monitor power and other stimuli applied to a UUT, but it should be noted that to connect an ATM pin directly to a PCM pin for this purpose will result in such an erroneous calibration. The correct procedure is to monitor at another point on the circuit under test.

On a 'bed-of-nails' this represents no problem; however an edge connector type jig may offer only single-pin access to PCM power supplies. These may still be monitored by either accepting measurement via an uncalibrated pin or, better, connecting an ATM pin via a relay (e.g. RRM) which is closed

under test program control. Note that if this latter solution is adopted, access to the ATM pin must still be provided for calibration.

The system ground pins are:

APG	Analog Power Ground
ASG	Analog Signal Ground
PSG	Pseudo Signal Ground
DGND	Digital Ground

Further information on ground connections is given in section 6.3.1.

Where more than one PSM* is plugged into the Test Jig, the source of the VREF calibration voltage, as already stated, will be the PSM of the highest address. However, for maximum convenience, it is best to wire both VREF outputs.

The major differences between the current supply module PSM and the superseded Power Control Module PCM are, in the first place, that when the power supply is not applied the PSM power rails have very high source impedance. In the second place, the Power Supply Module also allows a programmable current limit as well as, if fitted, the current limit facility involving a resistor.

* References to PSM also imply PCM.

5.0 TEST PROGRAM STORAGE

Test program storage is normally on mini floppy diskettes, each having a storage capacity of 72K bytes, in single density format.

The A8000 can also store programs on Erasable Programmable Read Only Memories (EPROMs) located on a Test Program Module (TPM) fitted to the underside of the test jig and connected to the program interface which, in this instance, is a physically linked pair of modules. This arrangement allows the A8000 to write a program to EPROM. A program stored on EPROM can be accessed and run by A8000 and by A8315.

During the writing of new programs, or modification to existing programs, the test statements are held in the system's RAM. As a precaution against loss caused by supply interruptions, it is advisable to transfer programs to disc - or to EPROM - as each ten or so statements are completed.

A program read out to the system RAM for modification (including the insertion or deletion of statements) can be written back to disc, when it will be automatically sorted and reduced to the smallest size. On A8000, it can subsequently also be written back to EPROM, without prior erasure, gradually filling spare space. Where space is at a premium, blank EPROMs can be fitted before the transfer is made. Storage is up to 16K bytes (eight 2K EPROMS).

5.1 DISC REFERENCING

The name of the appropriate program on a disc, to match a particular test jig, is indicated by a system of wire links on a connector on the jig and connected to the program interface in the rightmost slot. There are six groups of four pins each, relating to the six hexadecimal characters of the test program name.

Each

Each bit within a group is read as a '1' when linked to a ground pin (i.e. negative logic). Hence, a totally unlinked connector would reference a test program name of 000000 - for this reason, this is a prohibited test program name. The pin allocation of the program links is shown on page 5-2.

6.0 TEST JIG (TEST PLATFORM)

The Test Jig consists of a rigid insulated board with a metal skirt on three sides underneath. The board is drilled to carry the 'bed-of-nails' or other system for contacting the Unit Under Test (UUT). The right-hand side of the skirt may carry a Test Program Module (TPM) with its associated EPROM's. Alternatively, if disc program storage is used, it carries the wire-link connector (see section 5.1). In either case, connection to the main system is by a connector on the rear of the skirt, positioned to mate with the Program Interface in the Test Rack. Also on the rear of the skirt are the Test Interface Connectors, carrying supplies and test signals between the UUT and the appropriate test modules in the Test Rack.

In some circumstances it may be appropriate for a given test jig to be used with more than one program. To achieve this, disc program storage is used, and the wire links described under 'Disc Referencing' in section 5.1 are replaced by switched links. Typically, edge-wheel switches are mounted on the Test Jig to select any combination up to six hexadecimal characters.

During drilling of the test board it may have been necessary to remove the metal skirt. Great care must be taken to re-fit the skirt correctly to provide exact alignment of the test interface connectors with the corresponding test modules. A platform alignment jig is supplied by Wayne Kerr to facilitate this procedure.

6.1 TEST MODULES

The test modules provide the interface between the Unit Under Test (UUT) and the Test Bus. They plug into the Test Rack and make contact with the flexible bus at the rear of the rack. This flexibility allows the modules to be pulled forward to engage with the Test Interface Connectors (TIC) on the Test Jig, without losing contact with the Test Bus.

The test modules are not 'visible' to the system until they are addressed (Section 6.2), and so it is possible to leave test modules in the test rack even if they are not required by the test program.

The test modules can be situated in any slot in the test rack, except that the Power Supply Modules are restricted to the two leftmost slots because of their use of flying connector leads. In the absence of one or both PSMs

these slots may be used as normal. In addition, the Program Interface is restricted to the rightmost position in the Test Rack to enable the PLM to mate with the disc reference TIC or the TPM on the test jig. The Program Interface must always be present.

A list of the available Test Modules, together with a short description, is given below. Full specifications are documented in Appendix A.

<u>MODULE</u>	<u>ACRONYM</u>	<u>DESCRIPTION</u>
Analog Test Module	ATM	30 analog input pins
Power Supply Module	PSM	Dual prog 0-32V PSUs
Voltage/Current (I) Module	VIM	Dual prog DC sources
Sinewave Source Module	SSM	Programmable AC source
Reed Relay Module	RRM	24 relays for switching
Timer Interface Module	TIM	Real Time measurements
Impedance Test Module	ZTM	Tristate in-cct checks
Pulse Generator Module	PGM	Programmable pulse source
Shorts and Opens Module	SOM	39 continuity test pins
Logic Test Module	LTM	16 digital pins (I/O)
Digital Test Module	DTM	32 driver/sensor pins
Clock/Calendar Module*	CCM	Time/date records
Program Link Module	PLM	Program/Test Bus interface
Prom Enable Module	PEM	PLM servicing facilities
Customer Interface Module	CIM	Special UUT/System control
Power Control Module	PCM	Dual prog 0-32V PSUs (superseded by PSM)
Impedance Test Module	ITM	32 pins for in-circuit measurements (superseded by ZTM)

*Located (if fitted) in Control Box

6.2 TEST MODULE ADDRESSING

The systems derive their flexibility from the bus structure. This also allows any test module to be inserted into any position in the Test Rack, except for the two leftmost and the rightmost positions (see section 6.1). The systems have, however, to determine which test modules are to be used for a particular test program. As all Test Modules residing in the Test Rack are 'transparent' to the system until they are addressed by the user, it becomes a simple matter to address only those modules in the test rack that are required for a particular test program.

For the purpose of addressing the test modules, pins B9 to B15 inclusive are hardware address pins. These should be wired to address ground (pin B16) in a binary configuration, whereby the connection to ground makes that bit valid, to form the module address. The seven address pins allow the allocation of a unique address to each module, in the range 4 to 127, as shown in Fig 6.1. (Pin B9 = msb, pin B15 = lsb).

e.g. B13 linked to B15 linked to B16 = Address 5.

Addresses 1 to 3 inclusive are reserved for PSM's, which have restricted addressing with only pins B14, B15 and B16 being used. Once a module position has been defined by its unique address, it must thereafter remain in that position for the purpose of running that program. The system prevents the running of programs with incorrectly addressed test modules.

NOTE:

When an SSM is used as the AC stimulus for in-circuit testing, it is mandatory for the SSM to have an address of 4.

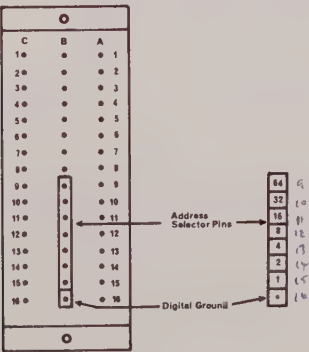


Fig 6.1 - Test Interface Connector (TIC) showing Binary Address Coding.

6.3 TEST JIG WIRING

6.3.1 Ground connections for functional testing

There are four internal ground systems:

APG	Analog Power Ground
ASG	Analog Signal Ground
PSG	Pseudo Signal Ground
DGND	Digital Ground

A fifth category exists, but is used only for address strapping.

In order to preserve measurement accuracy it is essential to avoid current flow in the Analog Reference (known as Analog Signal Ground, ASG) in the system and in the Unit Under Test (UUT).

Ignoring the ground used for Address Strapping, there should be a connection from each Functional Test Module Ground, where available, to the UUT ground. Ideally these should all be at the same point on the UUT but with a separate probe pin used for each. Where large circulating currents may be flowing between the system and UUT, this close spacing rule may have to be broken to prevent large currents flowing through the UUT ground, but normally a single good grounding point can be selected.

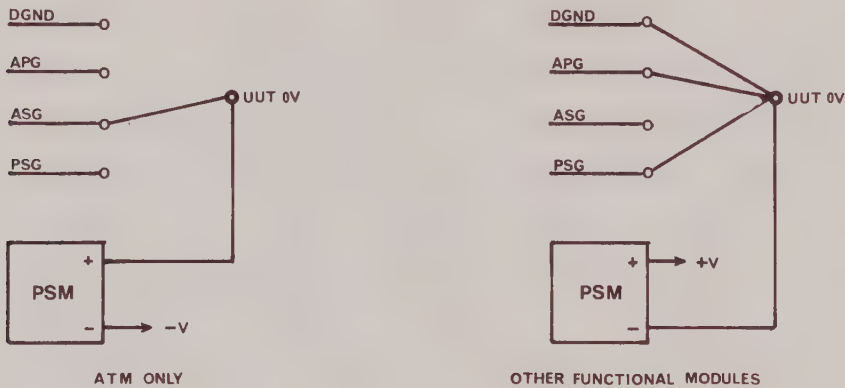
Note: The grounding requirement just described does not apply to the PSM, SOM, ZTM and RRM, which do not have UUT grounding pins. However, the floating PSMs, if used, must be referenced to the UUT ground.

The Analog Test Module (ATM) is the only one with ASG available to the UUT. When ATMs are the only modules providing a ground connection to the UUT, the real ASG (pin C16) should be used. However, if any other module requires a ground connection (e.g. LTM, VIM, SSM, PGM, TIM) then the Pseudo Signal Ground (PSG) of the ATM (pin A16) must be used if measurement accuracy is to be preserved. Provided that a ground connection is made in some way to the UUT (external or to the ATE), ATMs may be used fully differentially with no ground connection at all.

In a multiple ground configuration, the use of PSG avoids false current flow down the analog reference ground (ASG) which would otherwise reflect in impaired measurement accuracy.

In ATM-only installations (ignoring the excepted modules) it is permissible to use PSG on all except one ATM module, which should have a real ASG connection to the UUT.

Diagrammatically, the two situations may be represented as follows:



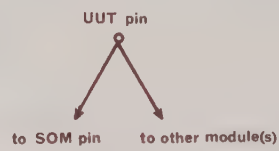
External connections Where connections to the UUT other than to the system exist, e.g. external power supplies or special measurement equipment, care must be taken.

An external ground connection to the UUT will cause a ground loop current to flow through the system Digital Reference which, under interference conditions, could result in operational errors and system collapse. In these situations it is recommended that the external equipment (or the system) be isolated by a transformer, or other means, where safety allows it.

6.3.2 (A) Wiring for SOM continuity testing

Each Shorts & Opens Module (SOM) has 39 pins for use in short-circuit and open-circuit testing. No special precautions are necessary in the wiring for continuity testing itself, other than ensuring that no external power supplies will be connected, since operation of the SOM involves a sequential grounding of pins on the UUT.

The system itself prevents operation of SOM continuity programs when functional/in-circuit modules are connected to the jig and vice versa. Hence SOM continuity pins can be wired direct to the UUT pin in parallel with any other wiring, as both relevant modules are never both connected to the jig at the same time.



6.3.2 (B) ZTMs for continuity testing (A8315 only)

Tristate Impedance Test Modules (ZTMs - see also next paragraph) can be used in A8315 solely for self-learned continuity tests, based on a known good unit, or for such tests in combination with functional tests which themselves may include in-circuit component checks. The ZTMs use optimising algorithms in learning continuity patterns, making testing very much faster. Operator instructions can be incorporated, short/open threshold values specified, and capacitor values input for the module to calculate delays to allow for charging times.

6.3.3 Wiring for in-circuit testing

Each Impedance Test Module (ZTM) has 32 pins which, under software control, can act individually in a tri-state Drive, Sense or Guard function. Provided sufficient access points to the UUT are installed and wired, guarding of subsequent component tests will be under program control.

When ac impedance checks are required on a component, a Sinewave Source Module (SSM) must be present, linked to address 4. Its dc-coupled output should be taken to the Stimulus Input pin (B1) of the appropriate ZTM(s).

6.3.4 Wiring for functional testing

In planning the wiring from stimulus and measurement modules to the Test Jig, the requirements for System Calibration (section 4.0) must be borne in mind. If ATM pins are wired through reed relays then, for calibration, a direct connection must be provided. This could take the form of a socket on the

test jig into which a shorting plug is inserted only for calibration purposes.

The calibration plate must not be allowed to contact any ATM pins connected to the + or - PCM outputs (regardless of whether one of these is or is not grounded). In the case of PSMs, contact with the + and - outputs is permissible ONLY after any ground link to either is broken. The isolation required (for PCM and ground connections) can be provided by relays or by drilling a clearance hole in the calibration plate so as to prevent contact with such psu pins or the ATM pins connected to them.

6.3.5 Wiring for combined in-circuit/functional testing using ITMs

An ITM pin cannot be switched off and, in the 'rest' state, is in the Guard mode, i.e. a virtual earth. Because of this, any permanent wiring of ITM pins to points which are subsequently used with 'power on' will affect the proper operation of the UUT with the possibility of damage to the ITM pin. To prevent this, all such ITM pins must be wired via a reed relay. Similarly, any Analog Test Module (ATM) pin wired to a point used for in-circuit driving will present an impedance of 1 megohm in parallel with the component being tested. More importantly, UUT power and ground must not be connected whilst these points are being used during in-circuit testing, and must therefore be wired via a reed relay. An ITM must always be declared as an ITM (see 6.3.6 for optional declaration of ZTMs).

Various configurations of reed relay wiring can be used to satisfy these requirements, the most common and useful probably being:



In many cases the number of in-circuit points will greatly exceed the number of points required for functional testing. In such cases the wiring shown above results in a 50% redundancy of the reed relay. Where this is critical, the arrangement shown below can be used:



IMPORTANT Since an in-circuit test point cannot be used simultaneously for more than one of the functions drive/sense/guard, then if the second wiring arrangement is used for ITM connections it must not be necessary to use both UUT nodes at the same time.

Other wiring configurations for particular applications can be developed. Please refer to Wayne Kerr for advice.

6.3.6 Combined in-circuit/functional testing using ZTMs

Used in A8315, the programmable tri-state (Drive/Sense/Guard) mode of ZTMs allows them to be used in a composite program embracing continuity testing (see 6.3.2 B), functional testing and in-circuit component checks. The use of isolating reed relays, as described in the previous paragraph for ITMs, is necessary with ZTMs only if voltages exceeding $\pm 15V$ will be applied to them. There may be a few other cases where relays are also required (see 9.5.15) otherwise, the ZTM pins may be wired in parallel with functional facilities on the same point.

Used in A8000, the functions of a ZTM do not extend to continuity testing.

A ZTM can be declared as a ZTM or as an ITM. However, declaring it as an ITM automatically switches all test pins into the GUARD mode and, once declared as an ITM, the tri-state mode cannot be invoked.

A ZTM used in A8315 and declared as an ITM, can be instructed to learn or test continuity patterns, automatically. ITMs and ZTMs (however declared) can be used together: however the test rack population (type of module and its address) must be the same for a test run as was used in the learning stage of programming.

6.3.7 Cross-talk

In those circuits under test where cross-talk or interference could occur, wiring on the test jig should be carried out using short 'point-to-point' runs and not be lengthened by tidying into neat cableforms.

7.0 TEST PROGRAM MODULE

The Test Program Module consists of a printed circuit board designed to hold the users program in EPROM. Up to eight EPROM's may be mounted on to this board to give a total program storage capacity of 16K bytes. A ribbon cable connects the TPM to the Test Interface Connector which in turn mates with the Program Link Module.

WARNING

On the TPM near the first prom socket is a red LED. Proms should not be inserted or extracted when this LED is illuminated. In some cases the VDU may offer the choice of a prom replacement when writing a program from A8000 to EPROM - that choice should be communicated to the system BEFORE proceeding with the replacement.

EPROM SPECIFICATION

Size:	16,384 bits
Organisation:	2048 bytes (2k x 8)
Power:	single +5V
Pin out:	As Intel 2716 or equivalent

NB NOT Texas 2716, but 2516

8.0 TEST PROGRAM GENERATORS

Upon detecting the operation of the PROGRAM key switch at a valid time, the system will switch into programming mode, and expect further operator commands and communication via the VDU. If no VDU is fitted, pressing RESET will return the system to the READY state.

The VDU, if present, will offer a choice of type of test program generator, either CONTINUITY or FUNCTIONAL. The former is for generating continuity programs using Shorts and Opens Modules. This is fully described in Sections 8.3 and 9.3 and uses automatic self-learning techniques from a known good unit. The facilities available are a subset of those provided by the Functional Program Generator; it is this latter generator whose facilities are described in Sections 8.1 and 8.2.

After choosing which generator is to be activated, the user may be prompted by the message "RE-ENTER GENERATOR". Replying a Y to this message will result in the respective generator being entered in a manner to recover a program previously resident within the generator. In some cases (e.g. incorrect exiting) the system may have erroneously returned to the READY mode, and this facility permits program recovery from this situation. Normally the reply to this would be an N (No), in which case the generator is entered in an initial state.

8.1 FUNCTIONAL PROGRAM GENERATOR (FPG)

Upon initial entry to the FPG, the user is presented with a program source choice:

NEW	EXISTING	DISC	RECEIVE
-----	----------	------	---------

NEW The FPG will request a program name. This should be a maximum of 6 hexadecimal (i.e. 0-9, A-F) characters. Name input can be terminated early with a <ret> at any time - in which case the name is zero filled, right justified.

EXISTING The FPG will read the program from the mini-disc unit or, if relevant, from EPROM. On successful loading, the program name will be displayed.

DISC The system displays the disc directory and invites input of the desired program name. Input of a program name not resident on the disc will cause "abcdef NOT FOUND" to be reported, and reversion to program source choice. <CTRL Q> as a disc name will also cause reversion to the program source choice.

RECEIVE The system will accept program source from an auxiliary SIO (RS232) interface configured as system port 2. The exchange is controlled by resident software protocol. The system will first output STX and then wait for six hexadecimal characters terminated by an ETX. Another STX will be output and the system will then expect a program statement followed by an ETX, up to a maximum of 126 characters. Each statement is validated on input - if an error is present, the system will present the statement for editing on the VDU. When it is successfully corrected, the system will recommence receiving from the SIO. This software handshake is shown diagrammatically in FIG 8.1. <CTRL Q> will terminate input at any time.

RECEIVER

i.e. System
in program
source 'R' mode

TRANSMITTER

i.e. System
in 'T' Function

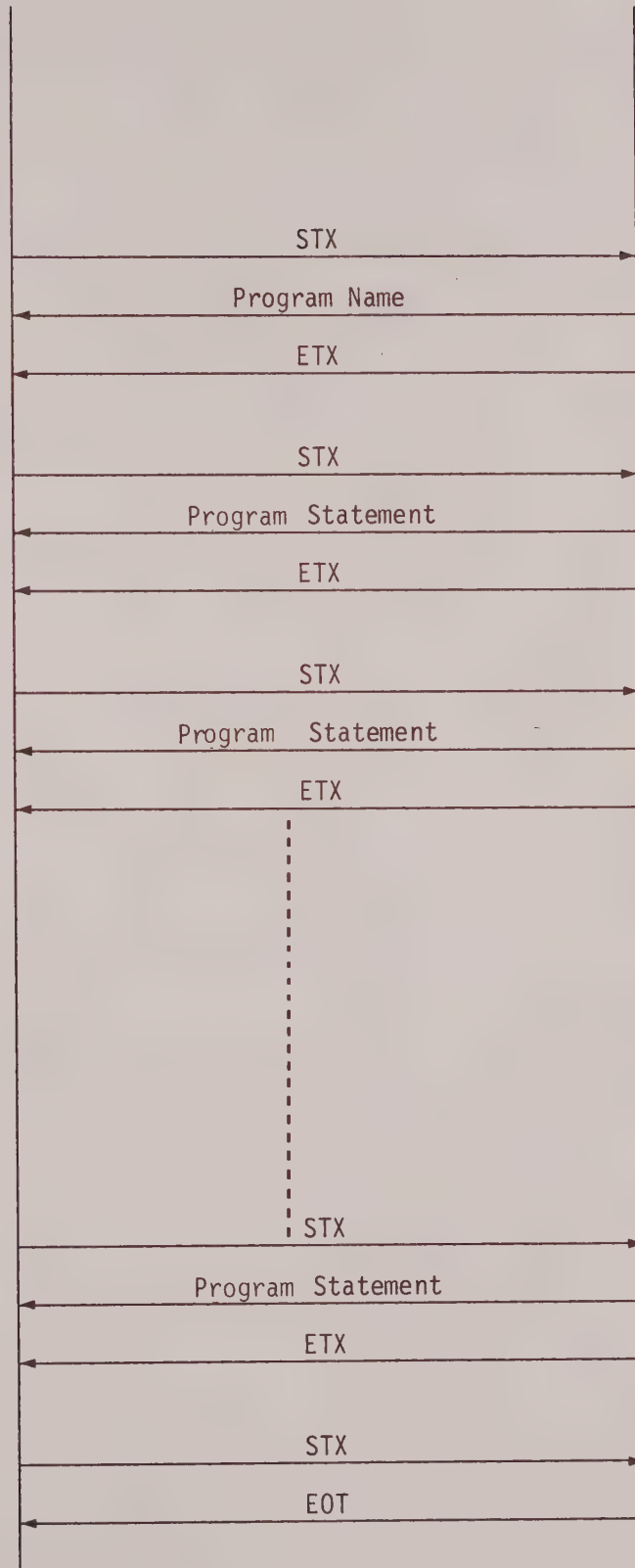


Fig 8.1 - Transmit/Receive Software Protocol

Subsequent to successful completion of any of the preceding four choices, the FPG will output the message "FUNCTION?", at which point it is awaiting user input.

The FPG contains a resident line and character editor which does not have to be explicitly activated - it implicitly operates whenever an FPG function detects the requirement for its use. This is fully described in Sec 8.2.

The FPG functions that are available are listed below.

FUNCTION?

A	Auto line numbering
C	Calibrate
D	Declarative mode
Fx	Floppy disc commands:
FC	Perform fast image copy from console drive to extension drive (if fitted)
FD	Delete file
FI	Initialise diskette
FL	List directory on VDU
FP	Print directory on ticket printer
FW	Write test program to disc
G	Guided probe
I	enter Interactive Program Generator (IPG)
K	Access to Calendar/Clock module (Time/Date entry, clock setting)
L	List program on VDU
M	Manual line numbering
NL	New Line numbering
NN	New program Name
O	Output (to port 2)
P	Print program on test console printer
R	Run test program
S	executable Statement mode
T	Transmit
W	Write to EPROM
X	eXit from FPG
Y	secondarY exit
Z	siZe of program
<CTRL Q>	Quit current function

These Functions are now described in more detail.

- A Switches to automatic line numbering. This allows the user to specify a statement number and the increment size. Subsequent statement modes (S D or I) will automatically use the current number, incrementing it each time a statement is completed. The System responds with the following prompt:
"AUTO LINE NUMBER/INC:"
<ret> causes the default values to be used.
n<ret> causes 'n' to be used as the initial value and the increment is defaulted to 10.
n/m will in addition cause the increment to be specified as m.
/m defaults the initial value and defines the increment as m.
The default values are:
- Initial value = 10
- Increment size = 10
- C Perform system calibration (see section 4.0).
- D Enter declarative statement input/edit mode. If not in automatic numbering mode, the user should input a statement number terminated by a space.
If a statement with the input number does exist, it will be displayed and the system will enter the line editor to permit modification. If a statement with this number does not exist, the user may enter a declarative statement (see Section 9.1).
A <ret> terminates input and, if syntactically correct, the system responds with a "D" on the next line to permit continuation of declarative input/edit. A <ret> with no statement number returns control to FUNCTION?. If the statement is not syntactically correct, the system will redisplay the line and enter the line editor to permit correction.
- Fx Disc control functions. All these are prefixed with an F and, unless otherwise stipulated, the commands act on the disc in the console drive unit.

- FC COPY DISC FUNCTION
- This command copies the contents of the disc in the console drive unit onto the disc in the Extension Disc Unit.
- Before typing FC, an initialised (but not necessarily 'empty') disc must be loaded into the Extension Disc Unit.
- The directory of this disc will be listed and the question "OVERWRITE THIS DISC?" asked.
- Typing Y (yes) will initiate the copy.
- The copy disc will have the identical disc 'name', since the process is an image copy.
- On the completion of a copy the system will Exit from the TPG.
- Warning! (1) Any previous information on the disc in the Extension Disc Unit will be erased.
(2) Any program resident in the TPG will be lost during the copy.
- FD DELETE PROGRAM FROM DISC
- The operator selects which program is to be deleted from the disc when asked "DISC PROGRAM NAME?".
- After entry of the name, confirmation is required.
- "DELETE abcdef?"
- Typing Y (yes) authorises the deletion of program abcdef.
- FI INITIALISE DISC
- A disc must be initialised before it can be used for the first time. The disc to be initialised is inserted into the console drive unit. Before Initialising begins, the message "INITIALISE DISC?" is displayed. Typing Y (yes) will authorise the Initialisation.

The operator assigns the disc a name when asked "NEW DISC NAME?". This name can be up to 8 alphanumeric characters long, and will remain unchanged unless the disc is re-Initialised or is used as the 'destination' disc in a Disc Copy,

WARNING

Initialisation erases any information previously stored on a disc.

FL

LIST DISC DIRECTORY

This command lists the program directory information of the disc on the VDU. This is automatically performed on the successful completion of commands FC, FD, FI and FW.

FP

PRINT DISC DIRECTORY

This command prints the program directory information of the disc on the ticket printer (if 'on').

FW

WRITE PROGRAM TO DISC

If a program with the same name is already on the disc, the operator will be asked for confirmation before writing.

"UPDATE EXISTING PROGRAM?"

Typing Y (yes) will authorise the program already on the disc to be overwritten.

G

Guided Probe. Enter digital diagnostic software packages. Refer to TP 217 (back of this manual).

I

Enter interactive program generator. A series of 'menu' choices or specific questions will guide the user in the construction of his program, without having to know the syntax details. More information is available in Section 9.9; for technical guidance reference should be made to the appropriate (Specifications) part of Section 9.9.

K

Clock/Calendar set up/modification sequence. For clock setting, the user is asked "INPUT YEAR?". Keyboard input may be four or two digits (e.g. 1983 or 83). An invalid input is reported as such. Pressing <ret> with no input causes default to year already stored on CCM. If none stored, message is:

"PREVIOUS SET UP INVALID" and user is requested to input a valid year entry. Next message is

"INPUT MONTH (1-12)?" Input, for example, 07 or 7 or default as for input year entry. Next message is

"INPUT DATE (1-31)?" Input must be compatible with month or INVALID will be output and user re-asked for date. This can only be defaulted if no new month was input. Successful operation brings

"INPUT TIME (HRS:MINS)?" Input required in 24-hour format. For single-digit hours or minutes, the leading zero can be omitted. If a time is entered, clock is set with seconds zeroed when <ret> is pressed. Time may be defaulted.

Successful entry of all clock parameters finally yields

"CLOCK SET TO"

19/11/83 14 : 05 00 (for example)

FUNCTION?

The clock has a four year calendar and will automatically allow for leap years. Further details of using the clock appear in section 9.9.22 and Appendix A.12.

L List the program on the VDU screen. The system responds with the following prompt:

"LIST LINES FROM/TO:"

<ret> causes the whole of the program to be listed.

n/<ret> will list all of the program starting from statement 'n' until the last statement.

n/m<ret> will list out statement starting at 'n' and finishing at 'm'.

/m<ret> will print from the beginning until 'm'.

<sp> will cause the listing to halt. Re-pressing <sp> will cause the halted listing to proceed from the point of interrupt. Addition of the suffix 'ALL' causes ZTM continuity programs to be full listed. Otherwise, the normal listing will include only the continuity command statement.

M Reverts to manual line numbering.

- NL New line numbering i.e. renumber. The first statement is renumbered as 10, and subsequent statements at increments of 10. The process automatically renumbers immediate (specified) destinations of GOTO, GOIF and EX statements, but cannot allow for destinations stipulated as a store content. The system displays 'renumbering' until complete. Renumbering will not apply to indirect addresses nor to undefined addresses which will be reported as invalid conditions.
NOTE: After renumbering, on A8000, it is not possible to 'overwrite' EPROM(s).
- NN New name i.e. rename program. The system displays the current name and invites input of the new one in the same manner as initial entry.
NOTE: After renaming, on A8000, it is not possible to 'overwrite' EPROM(s).
- O Sends the program out through SIO port 2, e.g. for printing on an external serial printer. Analogous to list.
- P Print the program on the console printer. The system responds with the following prompt:
"PRINT LINES FROM/TO:"
<ret> causes the whole of the program to be printed.
n/<ret> will print all of the program starting from statement 'n' until the last statement.
n/m<ret> will print out statements starting at 'n' and finishing at 'm'.
/m<ret> will print from the beginning until 'm'.
<sp> will cause the printing to halt. Re-pressing <sp> will cause the halted printing to proceed from the point of interrupt. Addition of the suffix 'ALL' causes ZTM continuity programs to be full listed. Otherwise, the normal listing will include only the continuity command statement.
- R Run program from RAM.
- S Enter executable statement input/edit mode.. If not in automatic numbering mode, the user should input a statement number terminated by a space. If a statement with the input number does exist, it will be displayed and the system will enter the line editor to permit

modification.

If a statement with this number does not exist, the user may enter an executable statement (see Section 9.2.). A <ret> terminates input and, if syntactically correct, the system responds with an "S" on the next line to permit continuation of executable input/edit. A <ret> with no statement number returns control to FUNCTION?. If the statement is not syntactically correct, the system will redisplay the line and enter the line editor to permit correction.

T As for O, but with the software protocol as described for Receive, the System acting as the transmitter. <CTRL Q> will terminate this action but will NOT send an EOT.

W Write program to EPROM. (Applicable to A8000 only).
If the program had been read in from EPROM, the system may offer the choice of 'overwriting' the existing EPROM(s), in which case THERE IS NO NEED TO ERASE THE EPROM(S) FIRST. Updating in this mannner creates redundant space within the test program. This redundant space can be 'freed' and the program compacted by choosing to use new (empty) EPROM(s). After the operation of functions NL and NN, it is mandatory to use new EPROM(s). The system performs much checking during this process, and may report comprehensively. Such dialogue is self-explanatory.

Note: If 'W' is used on A8315 the system will show "Write EPROM function not available" and return to "FUNCTION?"

X Exit from programming mode.
If a program resident in the FPG has not been written to EPROM or disc, the VDU will display "SAVE PROGRAM?" as a warning to the user. Replying N exits to the READY state; any other reply returns to "FUNCTION?". If, after exiting without saving the program, the user wishes to reaccess the program, it may be possible using the RE-ENTER capability (see Section 8.0).

Y Performs the same eXit, but re-enters the generator with the program source choice (New, Existing, Disc or Receive).

Z Causes the system to report the size of the program as number of disc tracks (equivalent to number of EPROMs).

<CTRL Q> (Control and Q pressed simultaneously)
Terminates current function and returns to FUNCTION?
Also cancels choice of Disc as program source, and reverts to the program source choice (see section 8.1).

8.2 FUNCTIONAL LINE EDITOR

The editor is implicitly invoked as described in the "S" and "D" function descriptions, and allows the deletion or modification of statements. The editor operates in two modes: COMMAND mode and INPUT mode. The COMMANDS available are:

<esc>	escape into command mode
A	Abort current edit
B	Begin edit again
nC	Change n characters
E	Echo line till End
I	enter INPUT mode
K	Kill (delete) characters (see description of nK on succeeding pages)
L	Look at line
T	Transfer characters (see description of nT on succeeding pages)
X	eXterminate (delete) line
n sp	step n characters
n bs	step backwards n characters
n rubout	rubout (delete) n characters

When editing, the "source" line is displayed above the "new" line. The process of editing involves selective transference of characters from the source line to the new line.

Below are full descriptions of these editing commands.

<esc>	Terminates current edit command and allows input of a new command.
A	(Abort) throw away edit lines and restart with original line as the source line.
B	(Begin again) Cancel existing attempt to edit and restart with the last completed edit as the source line.
nC	(Change) Replace the next n characters of the source line with the next n characters input.
E	(Echo) Copy text of source to the new line and enter the input mode at the end of the line.

- I (Insert) Enters the INPUT mode to insert text into the new line. The input sequence can only be terminated by an <esc>, <ret> or <CTRL Q>.
- nK (Kill) Skip transference of characters up to the n th occurrence of the character inserted after K. Used mainly in conjunction with Insert and/or Transfer commands. See Example at end of these descriptions.
- L (Look) Copy the rest of the source line to the new line, redisplay the line number and re-enter the edit mode.
- nT (Transfer) Copy from the source line to the new line up to and including the n th occurrence of the character inserted after T. Used mainly in conjunction with Insert and/or Kill commands. See Example at end of these descriptions.
- X (eXterminate) Skip transference of the remainder of the source line and enter input mode. If commanded when only the statement number of the new line is displayed, this acts as a statement delete function.
- n sp (space bar) Copies n characters from the source to the new line.
- n bs (back space) Operates only in the command mode, steps n characters backwards along the new line, erasing as it goes. Acts as the opposite of "sp".
- n rubout (rubout) Operates in INPUT and COMMAND modes
-Input mode - erases the last n characters input
-Command mode - skips transference of the next n characters BUT displays a "\$" character in their place in the new line.
- <ret> (return) Copies the rest of the source (as in command 'L'). Checks statement for syntactical correctness and reports any errors, allowing re-edit if errors are present.

Example of using I, K and T commands. With an original source line of:

S110 VER DCV, PINS 6/A1:C1, HL3.4, LL2.8

and a modified source line required to read:

S110 VER DCV, PIN7/C13, HL3.4, LL2.8

the sequence of commands would be:

- a) (1)TN (transfer characters up to and including the first occurrence of letter N. The numeral 1 is shown in parentheses since the system will assume 1 unless some other numeral is input).

This command will produce:

S110 VER DCV, PIN

- b) 2K1 (kill or skip transfer of characters up to the second occurrence of numeral 1 i.e. omit "S 6/A1:C1")
- c) I (insert)
- d) 7/C13 (required amended information)
- e) <ret> (carriage return).

This will produce the required new line:

S110 VER DCV, PIN7/C13, HL3.4, LL2.8

8.3 SOM CONTINUITY PROGRAM GENERATOR (CPG)

Upon initial entry to the CPG, the user is invited to select from

NEW EXISTING DISC

NEW The CPG will request a program name. This should be a maximum of 6 hexadecimal (i.e. 0-9, A-F) characters. Name input can be terminated early with a <ret> at any time - in which case the name is zero filled, right justified. Subsequent to this, the program generator will allow learning of the continuity test program as described in section 9.3.

EXISTING The CPG will read the program from the mini-disc unit or, if relevant, from EPROM. On successful loading, the program name will be displayed.

DISC The system displays the disc directory and invites input of the desired program name. Input of a program name not resident on the disc will cause "abcdef NOT FOUND" to be reported, and reversion to program source choice. <CTRL Q> as a disc name will also cause reversion to the program source choice.

Subsequently to successful completion of any of the preceding three choices, the CPG will output the message "FUNCTION?", at which point it is awaiting user input. The subset of functions available is:

FX	Floppy disc commands:
FC	Perform fast image copy from console drive to extension drive (if fitted)
FD	Delete file
FI	Initialise diskette
FL	List directory on VDU
FP	Print directory on ticket printer
FW	Write test program to disc
L	List
N	New name
O	Output
P	Print
R	Run
W	Write TO EPROM (A8000 only)
X	eXit

Y	secondary exit
Z	size of program
<CTRL Q>	Quit current function

For details of these functions, refer to the appropriately lettered sections of 8.1.

9.0 INTRODUCTION TO PROGRAMMING

9.0.1 Control Software

Whereas a test program contains instructions applicable to hardware functions, the term 'control software' is used to cover the additional instructions which determine the sequence or some other aspect of the test program as a whole. These 'control software' instructions have no effect on the test hardware: they govern the range of facilities available to the program - the architecture - that can be manipulated by the program instructions. The main facilities this relates to are Floating-point Numerical Stores, Test Flags, User Flags, Literal Store and String Variable Stores.

Floating-point Numerical Stores The machine has twenty floating-point stores, referred to as M0 to M19. The result of a test measurement is taken to one of these stores for subsequent use. They can also be used as instruction parameters. See Numerical Storage Capacity (page 9-2).

Test Flags The software provides six test flags. Setting either of the two main or global flags, PASS and FAIL, is obviously mutually exclusive. If FAIL is set during a test program, PASS will be unset. This change, which is unidirectional, is often caused by a check lying outside the limits specified for a Verify test. The state of these flags at the end of a test program provides the prime readout information to the operator.

GO and NOGO are two of four transitory flags determining the routing of a test sequence according to results obtained. GO is always set unless a check is failed. NOGO would be set when a measurement lay outside specified limits: otherwise it would be unset. The two are mutually exclusive and are truly bidirectional. The other two transitory flags, NOGOL0 and NOGOHI are extra qualifiers of NOGO. The four transitory flags provide the information for explicit fault diagnosis.

User Flags The machine has eight user flags, referred to as F0 to F7, which can be used to capture and store Boolean events, rather than absolute numbers. Additionally, some of these flags are used to communicate certain system events to the test program. F0, for example, would indicate when a time-out has occurred (e.g. it can be used to initiate branching to a diagnostic routine when no significant value

is obtained in the time allocated to a measurement). Further details on time-out are in section 9.0.5.

Flag F99 is set whenever a FAIL occurs but, unlike the FAIL flag, it is resettable. Thus it can be sensitised to record subsequent settings of the FAIL flag in a later section of the program.

Literal Store The machine has a twenty-character string store, referred to as M99. This can be used, for example, for input of a serial number and printing of that number on the 'ticket' for each operation of the test program. Since the information in M99 is in character form, it cannot be used for computation or as a system instruction parameter.

String Variable Stores The machine has five string-variable stores, N0 to N4, each capable of containing up to 80 characters. Their Numerical Storage Capacity is described in the next section. They can contain alphanumeric data or statements which may be called up, in whole or in part, for purposes of reference, comparison, identification or display. Their main function, however, is for IEEE command/data manipulation (see section 9.8).

There are three types of statement for string handling:

- SLET - assign characters to a store.
- SCOMP - compare two strings, setting the GO/NOGO flags.
- SLEN - put length of a string into a store.

In addition, the statements LET, IN, OUT, PRINT and DISP can make use of the strings. Further information on the String Variable Stores is in Section 9.9.20.

Numerical Storage Capacity The floating-point numerical stores (M0 to M19) and the string variable stores (N0 to N4) all accept numbers from 1×10^{-18} to 999.9×10^{15} which are stored in M0 to M19 as 32-bit floating point numbers. Whenever these numbers are entered, they may be in scientific notation, i.e. the range quoted above would be from 1E -18 to 999.9E15 (E = exponent). Statements OUT, PRINT, DISP and LET convert floating-point numbers, in this range, into an ASCII string. Within the overall range, the System will accept input statements, and provide readout, using these multiples and sub-multiples:

P : 10^{-12} N : 10^{-9} U : 10^{-6} M : 10^{-3} K : 10^3 MG : 10^6 G : 10^9

Computations taking numerical values into or out of this inner range will automatically cause the appropriate notation to appear.

9.0.2 Programming

A test Program is a sequence of events relating to circuit conditions under specified states of power input, signal stimulus, loading etc, such events being linked together in a particular way to provide overall information on the correctness of the wiring and performance of the circuit tested. Programming is the process of defining these events, in the required sequence, in such a manner that the machine will carry them out correctly. The process uses a language that is easily understandable by the writer of the program and unambiguously interpreted by the machine - according to certain standard rules - as specific instructions to the hardware. Before reading the description of the machine's Central Measurement System (section 9.0.3) or the general notes on program preparation (section 9.0.4) it will be helpful for the programmer to appreciate that statements put to the machine fall into two classes: declarative and executable.

Declarative statements define the required configuration of the test system. They will be utilised during the initial stages of a test sequence (while the machine is loading a test program) and are not actioned during the actual run time. Their purpose is to ensure that the hardware configuration is correct. For further information see section 9.1.

Executable statements make up the major part of a program, actively causing the hardware to perform specific functions. These statements can be classified according to the type of test to which they relate:

a) **VERify** - An instruction to check that a parameter conforms to some specified characteristics. In the case of functional tests, and for in-circuit checks on passive components, the characteristic will be defined by numerical high and low limits (HL and LL) set by the programmer. However, where checks are made on active components, in place of limits the requirement is that the non-linearity is similar to that simulated by the System. In the case of digital tests, VERify will be either TRUE or FALSE. (For further details see section 9.7.) In all the above situations, and with COMPFAIL checks, failure to conform results in FAIL being set, together with the NOGO and F99 Flags, (with passive components, the measured value from ZTM will be stored in M0 in all tests) and, if the printer is in use, the fail information will automatically

be printed. This indicates the need for deeper diagnosis or for program END to be called.

b) MEASure - For analog tests, the system is instructed to capture the data into a store for later use. If no store is specified, the data will be put into M0 (replacing any data already stored there). With digital tests, a MEASure is True or False. If False, only the NOGO flag is set, (FAIL and F99 are not affected) and this can be used within the program structure to create loops or formulate corrective procedures.

c) ADJust - The system is commanded to display to the operator a graphical representation of the physical/electrical setting of an adjustable control, where X's represent acceptable limits and 0 the latest setting. When the 0

X-----+--0---X

is coincident with X or +, an asterisk is shown. This display is a breakpoint in the test sequence, terminated by the operator pressing PROCEED. In other respects this mode is similar to VERify, and flags are set accordingly.

d) Relay - Tests requiring relays to be operated, such as the use of an RRM for load switching, signal routing, etc.

e) Stimuli - Instructions for connecting-in a programmable source and establishing required frequency, level, etc.

f) Timing - A 'timer' test commands the system to emulate a two-input counter/timer for "armed" timing measurements.

g) Digital - A class of test which permits the programmer to specify digital input and output conditions, and strobe (sample) points.

h) Continuity - The Tristate Impedance Test Module (ZTM), when used in A8315, allows self-learned continuity tests to form an integral part of a program also covering in-circuit and functional tests.

j) Control - Instructions to define the order in which various statements are executed, or to otherwise cause non-hardware effects. In this category is computation, and the instructions for COMPare and COMPFAIL operations.

Each class of test contains possibilities of several test statements,

every one of which has a unique meaning. Further information on executable statements is given in section 9.2, and detailed descriptions of the functions applicable to each are given in the appropriate later sections of this manual.

A program may have up to 1023 statements and each - whether Declarative (D) or Executable (S) - must have a unique number; i.e. the same number cannot be used for D and S statements.

9.0.3 Central Measurement System

Voltage/Time

The Central Measurement System, which is contained in the ACM and TRC, provides facilities for the measurement of voltage and time. D.C. voltage is measured using dual slope integration (a pre-set time is allocated to a ramp which ascends at a rate determined by the value of the unknown; this value is deduced by measuring the time for the controlled-slope descending ramp to return to the lower reference). Thus, fundamentally, all measurements are based on time. To minimise any errors due to supply-frequency (hum) pick-up, the integration time is usually set at a nominal 20ms (for 50Hz* supplies). For a.c. voltage measurements, an a.c./d.c. converter is interposed before the dual slope integrator, responding to the average value of the input. Hence the integration time for a.c. measurements must capture at least one complete cycle. The system computes and presents the measurement as an RMS value of a sinusoidal signal.

Input Circuit

To provide for flexibility in making the measurement connections to the UUT, the first stage of the measurement system has a differential input circuit. For subsequent ease of processing, the input on internal hi and lo bus lines is converted to a 'single-ended' voltage, the amplifier which makes this conversion serving also to provide eight alternative measurement ranges by means of programmed gain switching. The maximum input for each range is:

1.	0.5V	5.	8V
2.	1V	6.	16V
3.	2V	7.	32V
4.	4V	8.	64V

With ranges ascending in binary fashion, the worst case for accuracy is at half full scale and this is the accuracy figure quoted in the performance specification. The amplifier output is applied to an integrator via CT A - one of two counter-timer lines.

* Nominally 16.67ms for 60Hz supplies.

Range Selection

Selection of the appropriate range may be an implicit command derived from high or low limits programmed for voltage or current VERify or ADJust statements, or it may be an explicit command specified by the programmer. Autoranging will occur if no range command exists. This involves a first measurement on the highest range from which the appropriate range is selected for the final measurement (no second measurement being made for inputs between 32 and 64 volts). In general, testing time is reduced by specifying the range, since only one measurement is made rather than two. In such instances, any value exceeding the full scale capability of the selected range will cause "overrange" to be reported, while values lying below half full scale will be measured with poorer resolution and accuracy than is available from the system using a lower range.

Integration Time

In a similar manner, the programmer can select integration times other than the normal 20ms*. Examples of the need for such departures are: to ensure that measurements of low-frequency signals capture a whole number of cycles, or the choice of an integration time to minimise interference from frequencies other than 50Hz*. It should be noted that changes to the normal integration time must be considered in association with the voltage range selected. The maximum inputs specified above are for an integration time of 20ms: doubling the integration time on a given range will halve the maximum input voltage.

If autoranging is used, the system will compute the expected range from the input value and subsequently adjust to allow for the altered integration time. If both voltage range and integration time are specified by the programmer, without due consideration for their interdependence, the system may either report an error, or capture a value with reduced accuracy.

Where an integration time of less than 20ms* is specified, a lower range cannot be selected. Nominal full scale voltages yield 5V (average) at the integrator input. Overrange will be flagged if the ramp downtime exceeds 22.5ms, or if the integrator input exceeds 5.2V (average). This allows 'headroom' on nominal full scale.

* 20ms refers to 50Hz supplies: for 60Hz the figure is 16.67ms.

Timing Measurements

Two additional parameters must be specified when the Central Measurement System is to be used for timing measurements. These are threshold level and polarity. In the timing mode, the integrator circuit becomes a buffer amplifier and the input channel produces a digital representation of analog inputs when these pass the threshold level. To permit measurement on slow-transition signals, noise immunity of 10% of range full scale is automatically applied. Thus the applied signal must exceed the threshold level in both directions by at least 10% of the full scale of the range. The programmer can specify the range and the threshold level, giving the four operating conditions shown in the table.

Range specified	Threshold specified	Timing circuit conditions
YES	YES	D.C. coupled, threshold must be within range
NO	YES	D.C. coupled, range computed from threshold (to next highest)
YES	NO	A.C. coupled, threshold = average value
NO	NO	A.C. coupled, range 3, threshold = average value

In the event that range and threshold are both defined, but are incompatible (e.g. 2V range, 3V threshold), the system will report Error at Run time but it should be noted that the incompatibility may not be apparent before this. Specifying the threshold level has two advantages. Testing is faster, since no ac settling time is needed. Secondly, with signals having extreme duty ratios or mark/space ratios, with ac coupling the average (i.e. threshold) level may be too close to either excursion.

Polarity must be specified for pulse width and period timing, and for event counting, and may be specified for frequency counting, to provide start and stop actions from positive-going or negative-going edges. Event counting is performed within a programmed gate time.

Frequency Measurements

The usual method of measuring frequency - counting number of cycles in a given time - can be slow for automatic testing. W.K. systems operate by measuring the time occupied by a specified number of cycles (16 or 256). Thus, number of cycles is a parameter option available to the programmer. Where this is not specified, the system will first make a period measurement and compute the appropriate number of cycles to be measured in order to achieve the specified accuracy. (For frequencies below 200Hz the period measurement alone serves for the frequency measurement). Details are in Appendix A.1.

Other Functions

The Central Measurement System performs functions other than those just described (for example in Continuity and In-circuit testing) but in these situations it is controlled completely by the overall system and no programmer action is necessary.

9.0.4 General notes on drafting programs

Writing and implementing test programs is logical and straightforward once the correct sequence is adopted. The following notes will assist newcomers in preparing efficient and comprehensive test programs. It is valuable to consider the facilities available in six main groups:

Continuity testing - short-circuits and open-circuits

Verify checks - establishing that a parameter lies within pre-set limits or conforms to a particular characteristic

Adjustments - stages in the test sequence where pre-set controls on the UUT are to be adjusted

Measurements - values obtained from the UUT which may be displayed, printed-out, processed, or stored for later operations

In-circuit tests - measurements of component values using guard connections to neutralise the effect of adjoining components in the UUT

Functional tests - checks, adjustments and measurements made with a stimulus applied to the UUT

The test program can be written to achieve a GO/NOGO result or to provide information for fault diagnosis.

- 1 SOM continuity testing cannot form part of any other program. Should the same test jig be utilised for continuity tests and for functional tests, great care must be taken to ensure that only the appropriate modules for either one are connected to the test jig. With A8315, continuity testing using ZTM's can be incorporated into a functional test program.
- 2 If diagnostic testing is required, try to divide the circuit into simple sections that can be tested separately.
- 3 Decide on the parameters to be tested and the controls to be adjusted.
- 4 Decide on the requirements for stimuli and power supplies.
- 5 List the tests required, in sequence, including information on any stimuli, points of connection to the UUT and the type of measurement or check required. It is strongly recommended that a flow chart is compiled to aid program writing. (See the example in section 9.6.2).
- 6 From the foregoing, determine the quantity and types of test modules required (see section 6.1). Allocate unique addresses to these and specify their position in the test rack (slot number).
- 7 Specify positions of test pins and have the jig constructed (see section 6.0).

This completes the basic work : the test program itself can now be produced.

- 8 Declare the test modules used.
- 9 When a few test program statements have been entered, it is advisable to run this portion for verity and, when correct, to write these to disc before entering further statements, and keep back-up copies.
- 10 Run the complete program to check that it is performing correctly.
- 11 Correct and improve where necessary.
- 12 When satisfactory, write final version to disc and make backup copies.

9.0.5 Time Out

When measuring or verifying FREQUENCY, PULSE WIDTH, PERIOD or TIMER operations, the system relies on an input occurring within a given time. To prevent the system being 'hung up' should an input fail to materialise, a TIME-OUT will occur. The interval before this occurs is normally 12 seconds, but this can be extended (see below). Except in the case of IEEE control, a time-out causes F0 to be set. With the IEEE facility, if time-out occurs before an input commences, F0 is set; if it occurs before completion of an input, F1 is set. In all instances, when time-out occurs, the program should interrogate F0 to direct the system to a sub-routine, if necessary.

The effective length of the interval before time-out occurs can be extended, when time measurements are made using CTA and/or CTB, by delaying the start of the 12-second interval. An example follows:

```
S130 CON LTM, PIN 5/A6 ONTO CTA
S140 PREP TIMER MODE AA, START POS, STOP NEG
S150 DLY 8S
S160 EX S150 *2
S170 MEAS TIMER INTO M18
```

This would provide a total interval of $(3 \times 8) + 12$, i.e. 36 seconds before time-out. Should the condition have failed to occur at line S170, the equipment will wait for 12 seconds. If the condition has still failed to occur, the message NO PULSES - TIME OUT will be displayed and the next sequential line executed. This should be

```
S180 GOIF,F0,SXX
```

where SXX defines the start of a sub-routine.

9.1 DECLARATIVE STATEMENTS

Declarative statements define the presence of a particular test module in the test rack by allocating a particular address to the module. All such statements start with a 'D' followed by a unique number in the range 1-9999. Provided that they are input before an executable statement relating to that module (see section 9.2), they can occur anywhere in the test program. Test modules are transparent to the system until they are addressed and an attempt to write an executable statement to a non-declared module will cause the system to display UNDEFINED. In this event the statement must be terminated <CTRL Q>, the test module declared and the executable statement re-written. If a module is defined but is of the wrong type, the system reports INVALID.

There are fifteen slots in the test rack : the extreme right-hand slot is permanently populated by the Program Link Module (PLM). In the case of A8000, two slots are used. When one or two Power Supply Modules (PSMs) are present, they must occupy slots 1 and 2. Apart from these constraints, any test module can occupy any position in the test rack.

Examples of declarative statements are:

D1 ATM=4/6 Analog Test Module is allocated address 4 and is located in slot 6.
 D5 PSM=1 Power Supply Module is allocated address 1.

Omission of a slot number, as in the second example, can apply to any statement. Slot numbers are used only to provide part of the TEST RACK MISMATCH print-out and, consequently, their inclusion in a declarative statement is optional.

Due to the simple nature of these statements, the Interactive Program Generator does not provide facilities for the input of declarative statements.

Great care must be taken when editing declarative statements, since changes may invalidate previously input executable statements. Such changes will sometimes necessitate deletion and re-input of a statement to achieve modification.

Details of Test Module Addressing are in section 6.2 but it is emphasised that addresses 1 to 3 are reserved for PSMs and, when an SSM is used as the ac stimulus for in-circuit testing (i.e. with a ZTM), the SSM MUST be allocated the address '4'.

9.2 EXECUTABLE STATEMENTS

A classification of executable statements by type of test has been given in section 9.0.2. Executable statements instruct the system to provide control over the hardware and software. For instance SET UP, MEASURE and GOTO are all executable instructions in that they are used by the system as a sequence of operations.

All executable statements begin with an 'S', followed by a unique number in the range 1-9999 (but not using numbers allocated to declarative statements).

Examples of executable statements are:

```
S20 PRINT,W20, 'TEST N01'  
S30 MEAS DCV, PIN4/A1, INTO M16  
S40 COMP M16, HL1.5, LL1.2  
S50 GOIF, NOGO, S20  
S60 END
```

The language of test statements is detailed in section 9.9.

NOTE Programs are executed in statement numerical order (whether prefixed D or S) unless otherwise commanded by a jump. Lack of further statements automatically causes program termination : multi-branch programs should use the END statement to terminate branches.

IMPORTANT

In order to obtain full interactive information during programming, keep the printer switched on or ensure that S2 on S203 of the VDU/SIO port is set to give printer data shown automatically on the VDU. (See page B-1).

9.3 SOM CONTINUITY PROGRAMMING

Continuity tests can be made on loaded or unloaded boards and assemblies. Programs are constructed by a self-learning process, from a good board, using one or more Shorts and Opens Modules (SOMs). Because these modules present a low impedance current source/sense, it is essential that no other test module is addressed while SOMs are in use either for creating or running a test program. An SOM could be damaged by another module injecting current into it. For the same reason, boards for continuity testing (or programming) must not contain any charged capacitors. If a test module (other than an SOM) is plugged into the jig, the system will not allow programs to be either created or run.

In order that the system can distinguish between a real short-circuit and the apparent short-circuit initially presented by an uncharged capacitor, the user simply inputs the value of the largest capacitance between any two test points, and the system repeatedly inspects apparent shorts for a sufficient time for capacitors to charge and the correct (open or short) condition to be determined. By default, the system assumes the maximum capacitance present is 4 μ F.

The demarcation value of resistance between a short-circuit and an open-circuit can be established anywhere between 5 and 50 ohms, producing a linearly-related test voltage between 50 and 500 millivolts. Thus, even at the maximum setting, most semiconductors present a relatively high impedance. With no demarcation value input, the system defaults to 10 ohms. Default values are chosen by pressing <ret> without entering a value. The system will then prompt the user by displaying

SUPPLY OPERATOR INSTRUCTION

> <

If required, a 16-character user instruction (e.g. SET S7 TO RECORD) can be input: if no instruction is needed, pressing <ret> will cause the system to memorise the known unit.

The memorising process is a sequence in which the short or open result between every two pins is learned. Each SOM provides 39 pin connections (i.e. 39 test points) and up to 13 SOMs can be used at one time (a total of 507 test points).

If an operator instruction is input, the system will repeatedly prompt the programmer to input further instructions until <ret> is used without an instruction. The learning process then occurs.

If no operator instructions are supplied, the system will learn only one configuration of the UUT.

If operator instructions are supplied, on completion of the self-learning sequence for the first configuration of the UUT, the system prompts the user to supply instructions for another configuration. The cycle of learning for each configuration continues until the programmer uses <ret> in response to SUPPLY OPERATOR INSTRUCTION. On completion of the self-learning process for the last configuration, the program is complete. It can now be printed, listed, written to storage medium and run in the usual way. The subset of FUNCTIONS available (see also sections 8.0, 8.1 and 8.3) includes:

FX	Floppy disc commands:
FC	Perform fast image copy from console drive to extension drive (if fitted)
FD	Delete file
FI	Initialise diskette
FL	List directory on VDU
FP	Print directory on ticket printer
FW	Write test program to disc
L	List (program on VDU)
O	Output (program through SIO 2)
P	Print (program on test console printer)
R	Run (test program)
W	Write (to EPROM) (A8000 only)
X	eXit (from FPG)

NOTE. Programming with SOMs is exclusively a self-learning process: unlike functional test programs, editing is not possible nor can a program be created by any means other than self-learning from a known good unit.

When running a continuity test, any discrepancy between the UUT and the learned pattern will cause the system to print out an automatic fault signature such as:

O/C 4A1 - 5B6

(pin A1 on module 4 IS NOT connected to pin B6 on module 5 and IT SHOULD BE), or:

S/C 5A2 - 5A16

(pin A2 on module 5 IS connected to pin A16 on module 5 and IT SHOULD NOT BE).

IMPORTANT. Continuity programs learned exclusively on SOMs can be run on any of the systems (A8000, A8315 etc.). However, when SOMs are used, it is essential that the addresses used in testing runs are identical to those used in programming. With this proviso, a program established on one system (e.g. A8000) can be run on another system (e.g. A8315).

9.4 INTERACTIVE PROGRAMMING

The Interactive Program Generator (IPG) simplifies the creation of test program statements by presenting the user with the appropriate selection of choices at each step and by checking the syntax as each reply is input. The IPG is entered from the FUNCTION? mode by typing I. The user can return to the beginning of a statement by pressing <esc>.

Only the value of the asked-for parameter should be input: the key-word is supplied automatically by the IPG.

A question terminated with ! must be answered, while one terminated with ? indicates an optional parameter, where <ret> can be used to default. The remainder of the IPG dialogue is self-explanatory.

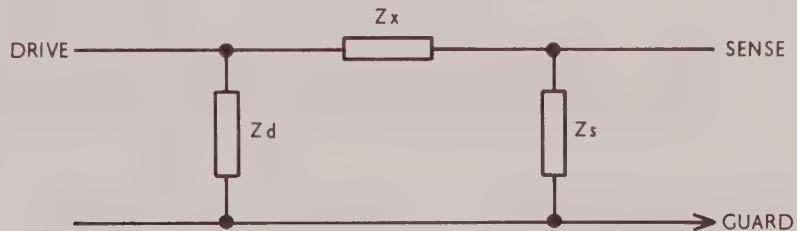
Use of <CTRL Q> at any time, or of <ret> when only a statement number has been requested, will cause return to FUNCTION?

9.5 IN-CIRCUIT PROGRAMMING

9.5.1 Introduction

Testing components already fitted into a circuit must take into account the shunting effects of other components in the circuit. It is convenient to consider these shunt elements as two lumped impedances, one each from the driven and sense ends, respectively, of the test component (Z_x), to a common point (see Fig. 9.1).

Fig. 9.1
Shunt Elements



The shunting effect of $Z_d + Z_s$ across Z_x can usually be reduced to a relatively insignificant level, by connecting the common point to Guard. With resistive values for Z_x , Z_d and Z_s , for example, measurement accuracy is not seriously impaired provided only that $R_d > R_x/1000$, $R_s > R_x/1000$ and $R_x, R_d, R_s > 3$ ohms.

Each Impedance Test Module provides 32 connection pins. The guarding just described is provided under program control when using the tristatable ZTM : when the two mode pins for Drive and Sense are specified for a particular component under test, all other pins in the configuration act either as Guards or are tristate (high impedance, approximately $5M\Omega$ to ground). In the more general case of complex impedances, and when maximum accuracy is essential, the impedances of the drive, sense and guard leads must be taken into account. This is detailed later (section 9.5.8).

9.5.2 Drive Levels and Sense Ranging

Because of the very wide range of possible impedance values to be checked, the drive level is a function of the high* limit (HL) programmed, and the sense range a function of the low* limit (LL), for each component (or effective impedance) to be tested. It is recommended, therefore, that the ratio of high to low limit does not exceed 10 to 1.

* For capacitors, since $X = 1/\omega C$, these two words should be transposed.

Where only one limit is specified, the system derives from this both drive level and sense range. For these reasons, the system provides for impedance verification rather than measurement.

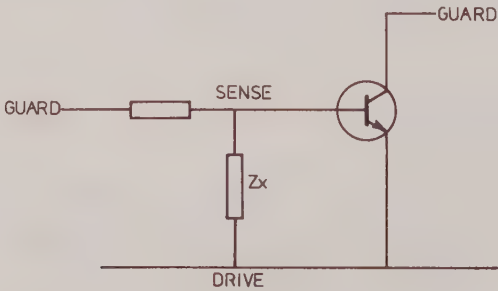
For resistance, capacitance and inductance, the drive level is determined by the highest value of impedance specified.

$Z_{max} < 1k\Omega$	:	5mV	rms (dc for resistance checks)
$Z_{max} \quad 1k\Omega \quad - \quad 1M\Omega$	:	5mV/k Ω	rms (dc for R checks)
$Z_{max} > 1M\Omega$	:	5V	rms (dc for R checks)

Note: Since $Z_c = 1/\omega C$, max. and min. impedance values correspond to min. and max. capacitance values, respectively. Maximum current is 5mA, regardless of test voltage. For resistors, system offsets are removed by performing the measurements with positive and negative drive voltages. For complex impedances, a sinewave centred on 0V is used. In a very few circuit configurations, where a high impedance is surrounded by relatively very low impedances, if these are passive it may be necessary to control any guard connection to them and, if they are active, to operate at a reduced stimulus level such that they are virtually non-conducting. See Reduced Level Testing, 9.5.15.

Where the shunt elements (Z_d and Z_s in Fig. 9.1) are of substantially different values, the connections to Z_x should be chosen so that the low impedance side is driven. This will ensure more repeatable results.

Where the shunt path includes a semiconductor, examine the circuit to assess the effect of changes in drive level. For example, in the circuit shown, the drive level to Z_x must not exceed the transistor V_{BE} .



To achieve a reduced level test (see also section 9.5.15) then drive levels could be:

for resistive Zx 0.3V dc (HL 60k Ω)
 " reactive Zx 0.2V rms (HL 40k Ω)

and value would be stored in M $\emptyset$.

For small transistors, 0.6V $V_{BE} \equiv 100\mu\text{A}$ base current, increasing at the rate of 60mV/current decade (e.g. base current 1mA $\equiv$ 660mV). The procedure is to specify the low limit (LL) for an artificially low Zx value, followed by COMPFAIL. Example, for Zx of 180k $\Omega \pm 5\%$:

```
S100 VER R, DR 7/A1, SN 7/A2, LL60K, CREF"ZX"
S110 COMPFAIL M $\emptyset$  HL189K, LL171K
```

For non-linear components (diodes, biodes and transistors), the drive and sense levels are derived from algorithms which model their electronic characteristics. See sections 9.5.9, 9.5.10 and 9.5.11.

Note: For any ac impedance tests, an SSM must be present and be allocated address 4. On the TIC's, the dc-coupled output (e.g. pin A3) of the SSM must be linked to pin B1 of any ZTM that will be used in a check using ac. There are 32 pins for impedance tests (A1 - A16 and C1 - C16). Seven of these (C2 - C8) can be used for single-pole connections or with double pole connections provided by B2 - B8. This can provide improved accuracy at low impedance, or better guarding in critical configurations. Any Impedance Test Module pins connected to power rails and very low value components should always have double pole connections which should be allocated before allotting other pins. The enhanced accuracy available by using double pole connections is detailed in section 9.5.8.

9.5.3 Resistance Checks

The ZTM checks resistance values from 3 ohms to 10 megohms with an unguarded accuracy of $(1 + 2R/10^6)\% \pm 0.12 \text{ ohm}$. If the measurement system detects a value less than 3 ohms, the result may be normalised to 0.0 ohms.

Since a dc stimulus is used, a delay must be introduced for capacitors in the guard path, or elsewhere, to charge. This is programmable between 100 microseconds and 8 seconds and must take into account such factors as the dielectric relaxation time of the electrolyte in certain types of capacitor, which can be several hundreds of milliseconds. It is worth noting that, because the dc applied to a resistor switches from positive to negative, a reduced level test

may be needed to ensure that electrolytic capacitors are not subjected to damaging reversed polarisation. This damage can arise at less than 0.5V reverse with electrolytics and at approx. 0.3V with tantalum bead capacitors.

9.5.4 Capacitance Checks

Unpolarised capacitors up to 1nF are checked at 1kHz($\pm 8\%$) and 1nF-8 μ F at 150Hz. The range is 50pF to 8 μ F with an unguarded accuracy of $\pm(2\% + 20\text{pF})$.

9.5.5 Electrolytic Capacitors

These are checked at different frequencies, depending on their value.

At 100Hz($\pm 8\%$) : 0.1 μ F - 160 μ F.	} Unguarded accuracy: $\pm 2\%$
At 20Hz($\pm 1\%$) : 160 μ F - 800 μ F.	

Since there are no totally reliable methods for non-destructively testing correct orientation, no polarising voltage is applied.

9.5.6 Inductance Checks

Inductors are checked at 200Hz($\pm 8\%$). The range is 8mH to 8kH, with an unguarded accuracy of $(2 + L/400)\%$. Care must be taken when checking inductors, due to the relatively high resistive component (especially so if the component is designed for use at frequencies much higher than the test frequency of 200Hz). It is recommended that in such cases an inductance check should be made in conjunction with a resistance check.

9.5.7 Impedance Checks

In addition to the case just described, there are other circuit configurations (e.g. a resistor and capacitor in parallel) where full component verification demands two checks, such as resistance and impedance. To allow optimum discrimination, the frequency used for the impedance check is programmable over the range 20Hz to 200kHz (accuracy 0.1%). For accuracy, it is best to employ as low a frequency as possible, compatible with the requirements for component type identification.

Complex impedances in the range 10 ohms to 1 megohm are checked with an unguarded accuracy of:

$2\% + (0.75\% \text{ for every } 100\text{Hz above } 300\text{Hz})$

e.g. 22% at 3kHz.

9.5.8 Accuracy of R,C, L & Z

In addition to the previously specified accuracies, further errors may result from the effect of lead impedances. On the drive and sense pins, these will normally only be important when checking low impedances; on guard pins, the effect will be determined by the ratio of the unknown to the shunt impedances.

These errors may be considerably reduced by using pins which have the double pole capability, by wiring the double pole pin to connect with its respective measurement pin as near the component lead as possible. If a double pole pin is not wired, then the respective measurement pin will operate as a normal single pole pin. See Appendix A7.

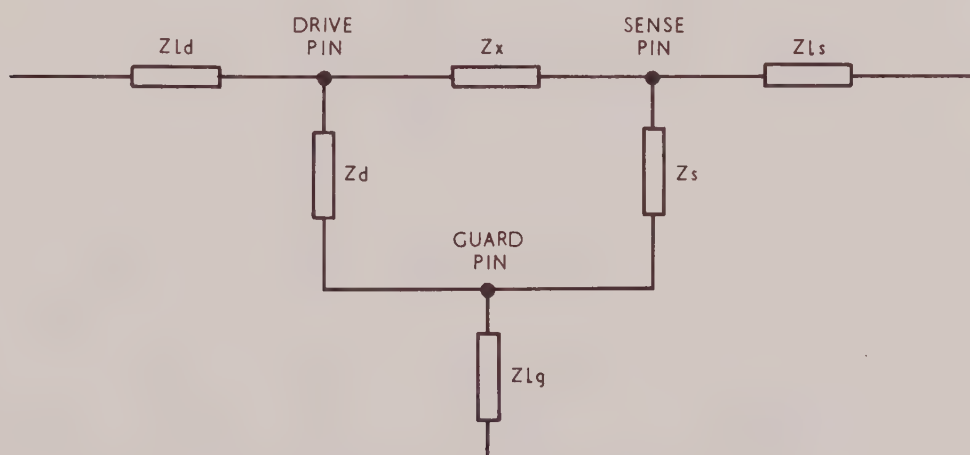


Fig. 9.2 Lead Impedances

Typically, drive source impedance and sense sink impedance are each less than 0.15Ω , and the line impedances each side of Z_x are about 0.1Ω , giving Z_{ld} and Z_{ls} (Fig. 9.2) as each approximately 0.25Ω . Double-pole connection to Z_x reduces these to effectively less than 0.015Ω each. Thus, considering only these series elements, if Z_x is 10Ω , single-pole error will be 5%, reducing to 0.3% with double-pole connections. The errors will be correspondingly smaller for Z_x values above 10Ω .

Taking into account the additional error caused by shunt elements Z_d and Z_s , and the lead impedance of the Guard connection Z_{lg} , the percentage error with

only single-pole connections is given by the equation:

$$\text{Error} = \left(\frac{Z_{ld}}{Z_x} + \frac{Z_{ls}}{Z_x} + \frac{Z_{ld}}{Z_d} + \frac{Z_{ls}}{Z_s} - \frac{Z_{lg}}{Z_d + Z_s} \cdot \frac{Z_x}{Z_d + Z_s} \right) \times 100\%$$

For single pole pins	$Z_{ld}, Z_{ls}, Z_{lg} < 0.15\Omega$	} +0.001Ω/Hz
For double pole pins	$Z_{ld}, Z_{ls}, Z_{lg} < 0.015\Omega$	

The calculated error will indicate whether or not double-pole connections to the Drive, Sense and Guard pins are necessary. Double-pole connection to the Guard node is only important when the shunt loads (Z_d, Z_s) are low in value and the ratio of Z_x to Z_{shunt} is large. Typical values for the series impedances quoted above are dc values and, because they are in fact frequency-dependent (see Appendix A.7), the accuracy must be calculated for each particular circuit configuration.

The measurement system will not work reliably if either:

$$\begin{aligned} &Z_d < \frac{Z_x}{1000} \text{ or } < 3\Omega \quad \text{or} \quad Z_s < \frac{Z_x}{1000} \text{ or } < 3\Omega \\ &\text{or } Z_x < 3\Omega \end{aligned}$$

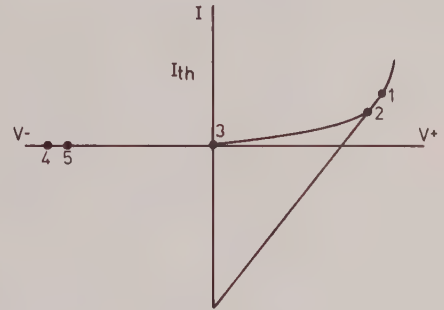
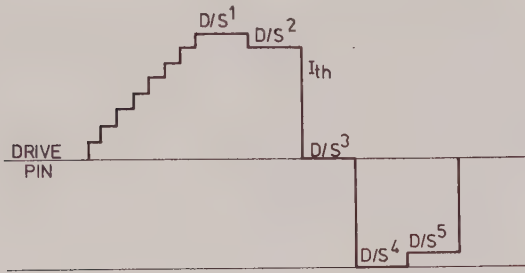
In these circumstances, provision must be made to reduce drive levels * (see section 9.5.15).

9.5.9 Diode Testing

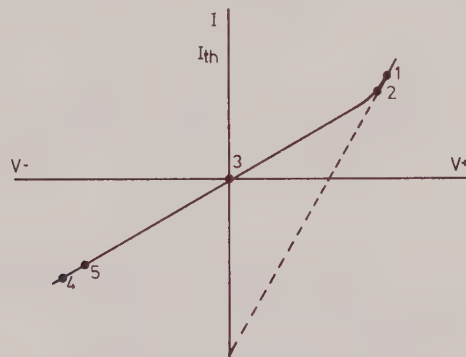
Wayne Kerr ATE systems are capable of checking diodes for correct polarity and failure to follow a model voltage/current path. The software uses algorithms which detect the non-linear component of a characteristic and, hence, can successfully test diodes even when shunted by resistance as low as 500 ohms. Diodes testable include germanium and silicon types, zener diodes and light-emitting diodes. Tests are performed under controlled conditions: in-circuit test modules supply a maximum of 5V dc to the component, but will terminate the test (reporting either pass or fail) if the device current exceeds 1mA.

In essence, a DC voltage is applied in the forward direction to the diode and stepped upwards until conduction exceeds 1mA. The drive voltage is then decreased by a small amount, and the conduction current re-measured. From these results, the non-linear current is deduced and checked. The same voltages are then applied in the reverse direction, and the procedure repeated to check

low reverse current in the diode. The sequence of drive voltage stepping, the associated sense measurements and corresponding characteristic are shown below.



With parallel-connected diode and resistor the V/I curve will depend on the R value but will generally be as shown:



Provided that the extrapolated diode slope does not approach the origin (point 3), the diode will be declared good. With smaller and smaller parallel resistor values the slope of the characteristic becomes steeper until the extrapolated diode slope approaches the origin. In this situation, the system will report an S/C or resistive failure.

Where the diode is associated with an impedance of less than 500Ω to ground (guard) on the driven side, although the slope may not be steep, the degree of non-linearity will be so small that once again the extrapolated diode slope will pass very close to the origin. In this instance the failure report will be O/C or resistive.

Where a circuit includes large-value capacitors, it may be necessary to introduce a delay before diodes are tested. The procedure is described in section 9.5.16 (Delays for Active Tests).

Associated components, (e.g. electrolytic capacitors), may impose restrictions on the polarity of the applied DC. To allow for this, there are two programmable modes for diode testing:

- a) Drive Voltage Anode (DVA) : The cathode pin is sensed at virtual earth, and the anode is driven positive and negative.
- b) Drive Voltage Cathode (DVK) : The anode pin is sensed at virtual earth, and the cathode is driven negative and positive.

The algorithms used for diode testing will perform correctly for any device which conducts 1mA forwards at less than 5V. However, to avoid other component damage in the case of incorrect assembly, the maximum forward voltage is programmable. In the absence of the parameter in the statement, the maximum voltage will default to 1 volt.

In the presence of a fault, the system will attempt to characterise the component into the following categories:

REVERSED

S/C OR RESISTIVE - high forward current, non-linearity small

O/C OR RESISTIVE - undetectable forward current.

If unable to do this, the diode will be reported as FAULTY.

Zener diodes will almost invariably pass as ordinary diodes provided VMAX is not raised above 1V. But they can also be further verified by testing them as a Biode with a VMAX above their breakdown voltage. So up to 4.7 volts should be testable.

Light-emitting diodes (LEDs) have characteristics similar to silicon diodes, except that the forward conduction point occurs at about 1.6V, so tests can be made by raising the VMAX field to 2V.

Note. The drive pin always stimulates with both a positive and a negative voltage unless a short circuit is detected. The presence of diodes in a circuit may necessitate reduced level testing for resistors and reactive components. See section 9.5.15.

9.5.10 Biodes

This term is used to describe a special combination of two components that cannot be tested separately, namely a pair of reverse-connected parallel diodes.



Such a combination will be checked for diodic forward conduction in both directions. Due to its symmetrical nature, the nodal connections are described as a drive pin and a sense pin. The drive pin is taken positive and negative to check both directions of conduction. As with a single diode, a delay can be introduced (see 9.5.16) and a maximum searching voltage can be specified.

If not correct, the system will check to see if the combination fits the 'O/C OR RESISTIVE' class of failure and, if it doesn't, will simply report it as FAULTY.

9.5.11 Transistor Testing

Bipolar transistors are tested in the common base mode. The collector or emitter is driven, the pin being sensed is a virtual earth, and the third terminal is guarded and therefore also at a virtual earth, as is usual for all other ZTM pins. A forward conduction test is made by driving the emitter, guarding the base and sensing the collector, the emitter drive polarity being dependent upon the transistor type (i.e. PNP or NPN). The collector current I_C is characterised for its non-linear component $I_{C(NL)}$. Then the collector is guarded, the base sensed, and the non-linear base current obtained $I_{B(NL)}$. The forward gain (FG) is thus obtained as $[I_{C(NL)}]/[I_{B(NL)}]$ and the device is failed if FG is too low. However, if $I_{B(NL)}$ is unresolvable due to shunt paths between the transistor terminals, or very high gain, a fixed value for $I_{B(NL)}$ of $8\mu A$ is used in the equation, resulting in what is called 'minimum' forward gain (as opposed to 'real' forward gain), which is not tested at this stage.

In a similar manner, a reverse conduction test is made by driving the collector, sensing the emitter etc., and a figure for the reverse gain (RG) is obtained, which may or may not be a 'minimum' reverse gain.

If forward gain is real the device is failed if $FG < RG$.

If both gains are minimum the device is failed if RG is irrational.

If forward gain is minimum and reverse gain is real the device is failed if

FG<RG and RG>25. In the 'print all' mode, the actual gain figures are printed, together with 'real' or 'minimum'.

Darlington pairs must be tested as two diode configurations.

The procedure for introducing a delay, to allow for large-value capacitors to charge, is described in section 9.5.16.

9.5.12 Transistor Error Reporting

A transistor may be classified as faulty as follows:

NO FWD I_C	i.e. no collector current was detected in forward gain test.
FWD I_C LIN	i.e. no non-linear current component was detected in the forward gain test e.g. high shunt path between emitter and collector.
NO REV I_E	i.e. no emitter current was detected in the reverse gain test.
REV I_E LIN	i.e. no non-linear current component was detected in the reverse gain test.

If the transistor is classified as faulty due to the gain tests failing, the values of both gains are reported.

9.5.13 Single Direction Transistor Testing

In some circuit configurations, the full transistor test described above may be too severe. In such cases the use of the defaultable parameter

ONLY FWD

ONLY REV

causes only the forward gain test, or the reverse gain test, respectively, to be made. An example would be an array of transistors with their emitters coupled together. With all bases guarded, the ZTM pin does not have sufficient drive capability to operate all the transistors in parallel. However, the reverse gain test, driving the collectors individually, will operate.

The gain tests for unidirectional testing are different since the test FG>RG cannot be made.

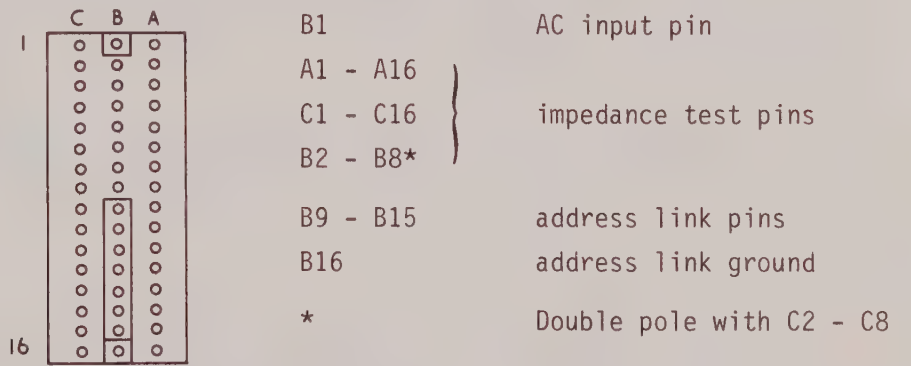
If ONLY FWD is specified a 'real' forward gain is tested for >15, and a 'minimum' forward gain is not tested, although it will be reported in 'print all' mode.

If ONLY REV is specified, 'real' or 'minimum' reverse gain is tested for <25.

In cases where even unidirectional testing is not possible, the transistor should be tested as a diode model, using two VER D statements.

9.5.14 ZTM Connections

As already stated in 9.5.2, when testing components requiring an a.c. stimulus, an SSM must be present, with address 4, and (on the TIC's) its d.c.-coupled output (e.g. pin A3) must be linked to pin B1 or any ZTM involved in these tests.



9.5.15 Reduced Level Testing

In cases where the stimulus level is too high for the associated circuit, the impedance of the component can be measured using a reduced drive voltage. An example of this requirement is often where components surround an integrated circuit. It is not possible to guard out the parallel paths within the IC since access to the internal nodes is not provided. Reducing the drive voltage will 'turn off' the IC parallel path, and enable measurement of the external component alone.

Reduced level testing is possible because all in-circuit VERify statements for passive components also deposit the actual value recorded in MØ. Since the stimulus level is derived from the low limit when the high limit is defaulted, a VERify statement may be specified with an artificially low 'low limit' to reduce the stimulus level. It is then followed by a COMPFAIL on MØ against the true component limits.

EXAMPLE:

A resistor of $120K \pm 5\%$ is across two pins of an IC which 'turns-on' above 0.3V.
The normal statement:

VER R,DRx/by,SNx/by,HL126K,LL114K,CREF ref

would result in a drive voltage of $126 \times 5mV = 630mV$.

However, the statements:

VER R,DRx/by,SNx/by,LL50K,CREF ref.....(1)

COMPFAIL,MØ,HL126K,LL114K

results in the resistor being measured with a drive voltage of $50 \times 5mV = 250mV$.

The VERify statement will pass, and the actual value will be checked by the subsequent COMPFAIL statement.

If, instead of resistance, the test is of capacitance, inductance or impedance, the drive voltage is ac and the peak value must be taken into account when specifying the rms voltage, e.g. $250mV\text{ pk} = 177mV\text{ rms}$. So, in equation (1) above, the statement would be

VER Z, FREQ 200.00HZ, ... LL35K,CREF.....

Reduced level testing will decrease the accuracy of measurement. The amount by which it will be decreased is very complex, and will depend on the value of a number of parameters. In general the formula

$$\text{Error (\%)} = 0.2 \times \frac{Z_x}{Z_{LL}}$$

will yield an approximate figure for the additional accuracy component. Hence, for the earlier example, the additional error would be

$$0.2 \times \frac{120K}{50K} = 0.48\%$$

It is wise to verify that the system is performing to the required accuracy at program development time. If required, Wayne Kerr can provide a detailed accuracy expression for a particular test situation.

Notes :

- 1) In the inverse manner to that described above, the system can be used with increased level of drive stimulus to enhance accuracy, by defaulting the Low Limit. It is not possible to characterise accuracy improvement in any meaningful way, so it is necessary to verify the accuracy improvement against a known value at program development time.
- 2) Maximum guaranteed ZTM drive current is 5mA, (see 9.5.2) hence avoid empirically generated programs, etc.
- 3) As a general guide, use Guarding wherever possible (this is easier on small boards) but, because of 2), if the shunt impedances are less than 3 ohms, or less than $Z_x/1000$, use tri-state.
- 4) When a test point has ZTM and LTM or DTM connections in parallel, components cannot be measured if the point is driven more than 0.5V negative on any in-circuit test. In these situations, either the other end of the component must be driven, or a relay must be interposed between the LTM/DTM pin and the UUT node.

9.5.16 Delays for Active Tests

A defaultable delay can be included in statements for testing diodes, biodes and transistors, to allow for large-value capacitors associated with these non-linear devices to charge before their characteristics are checked.

Examples are:

VER D, DVA, AN8/A1, KA8/A2, MAXV2.0V, DLY0.5S, CREF"D1"

VER B, DR8/A3, SN8/A4, MAXV1.5V, DLY0.35S, CREF"B1"

VER NPN, E8/A5, B8/A6, C8/A7, DLY0.45S, CREF"TR1"

If DLY is defaulted, there is zero delay.

9.5.17 ZTM Application

Declaratives and Default conditions

When declared as a ZTM, the ZTM defaults at program start to Tristate on all pins.

If declared as an ITM, the ZTM defaults at program start to Guard on all pins.

Stimuli Removal after Functional Testing

Before using the ZTM for in-circuit or continuity testing, ALL stimuli or power sources must be removed and capacitors allowed to discharge. To aid this process, use the REM ALL statement followed by a DLY. (Note: A8000 systems must REM each individual stimulus channel discretely, there is no REM All). As a safeguard, the continuity statements automatically re-initialise all stimuli modules but cannot insert any delay. Any external stimuli must be cleared down discretely, i.e. IEEE or RS232 controlled sources.

Stimuli Re-application

It is essential that the ZTM pins are not left in Guard state on resumption of functional testing. It is therefore good practice after in-circuit testing to use the TRI ALL statement.

Following in-circuit statements, APP statements will return stimuli modules to their previously programmed values.

Following continuity statements, all stimuli modules must be re-programmed from scratch. This should be straightforward by using EXecutes of the original setting-up statements.

Maximum Voltages in the presence of ZTM pins

It is imperative that voltages greater than 15 volts (+ or -) away from system ground (APG) are not allowed to reach the ZTM pins even while TRIstate. Failure to observe this rule may result in destruction of the module. Care must be taken, when a voltage greater than this is used on a board, that failure paths cannot create the forbidden condition.

Continuity and Declarative

Even if a ZTM is declared as an ITM it will be included in any continuity test. Therefore, great care must be taken if mixing ITM and ZTM modules in programs including continuity statements.

Guard States during In-circuit Measurements

When declared normally, the ZTM will remain in TRIstate unless explicitly commanded to be Guard, either as individual pins, whole modules or ALL. Therefore the Guard state of the pins at any point in the program is the result of the explicit GRD and TRI program statements. The only exception

to this rule is that CONTINUITY statements leave all ZTMs in the initialisation state appropriate to their declaratives, i.e. normally TRiState.

It is not necessary to explicitly Guard the pins before an in-circuit statement as this is handled by the system and all pins are left as found.

Continuity Failure Reporting of S/C Strings - A8315

Most continuity failure print-outs are straightforward. However, when a S/C string is greater than two pins long, some confusion may occur unless it is realised that the first pin in the string is tested against each other pin in the list. The fail reporting then becomes clear. Consider learnt S/C string:

Q10 (100) 5/A1 : 5/A2 : 5/A3 : 5/A4

If the continuity breaks between test points monitored by A2 and A3, then A3 and A4 are still linked. The Fail report will be:

OPEN CIRCUIT PAIRS

5/A1 - 5/A3

5/A1 - 5/A4

SHORT CIRCUIT PAIRS

5/A3 - 5/A4

confirming that A3 and A4 are still connected. If the short-circuit message does not appear, then A3 and A4 are independently disconnected from A1:A2.

If a set of short circuits to ground are learnt, they are listed as separate Q statements. But if the common ground fails we get the report:

OPEN CIRCUIT PAIRS

5/A5 - GND

5/A6 - GND

SHORT CIRCUIT PAIRS

5/A5 - 5/A6

which again is explicit.

A short-circuit string failing short-circuit to ground will only report the first pin of the string as short circuit to GND, all others being implied by the original string being found intact.

9.5.18 In-Circuit Testing of Transformers

The system with Impedance Test Modules (ZTMs) can be used to verify the correct assembly of transformers onto PCBs.

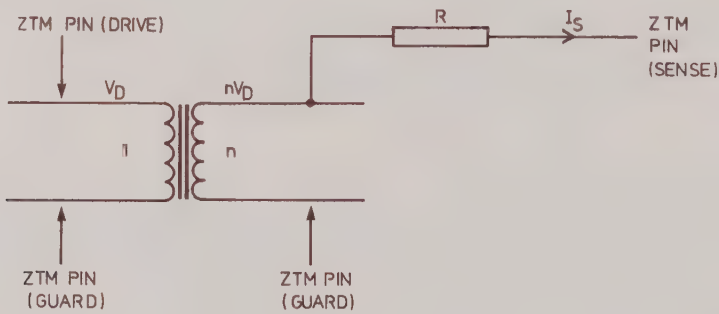
The two parameters that are important at assembly time are:

- i turns ratio
- ii phasing

It is expected that the other transformer parameters (e.g. leakage, saturation current) have been controlled by quality procedures during design and procurement.

1 Turns Ratio

Checking the turns ratio of a transformer will verify that, for example, the correct type has been installed, and that there is no primary/secondary cross. The technique involves mounting a resistor on the test jig, connected between an ITM pin and one end of the secondary winding, viz.



Both guard pins and the sense pin present a virtual earth. One end of the primary winding is specified as the drive pin; during an impedance test, an AC voltage of magnitude V_D is applied to this point, of frequency as specified in the test statement.

The system is effectively checking impedances by measuring V_D , I_S and calculating Z as $\frac{V_D}{I_S}$.

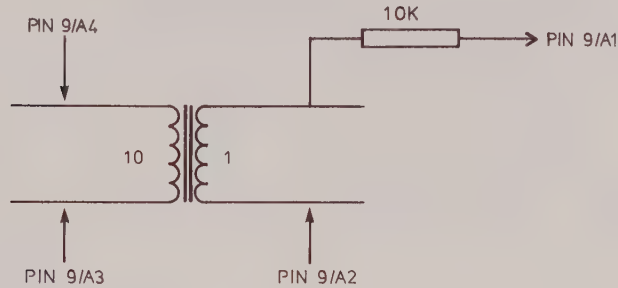
Since the sense point is a virtual earth, $I_S = \frac{nV_D}{R}$ and hence the

impedance that the system will 'measure'

$$Z = \frac{V_D}{I_S} = \frac{R}{n}.$$

Note: If the transformer is a step-down, and therefore normally expressed as an $n:1$ ratio, the result will have n inverted. i.e. $Z = nR$

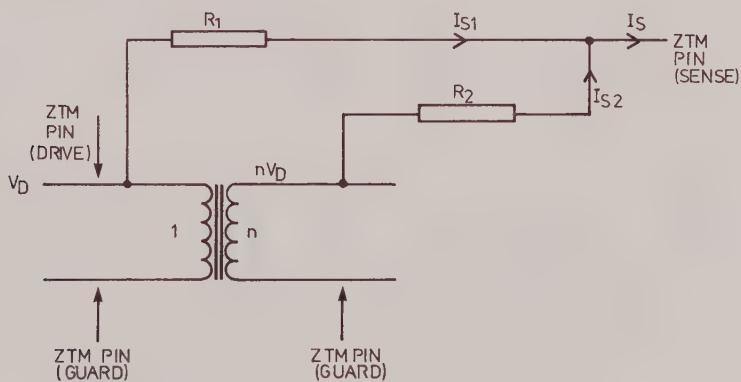
Example



To test this, the statement
`VER Z,FREQ50HZ,DR9/A4,SN9/A1,HL110K,LL90K,CREF"T1"`
 will check the turns ratio to $\pm 10\%$.

2 Turns Ratio/Phasing

In some applications, it is also useful to check that the transformer windings are connected to the PCB in the correct phase relationship. In a similar manner to that described above, this can be accomplished using two resistors.



Since the sense point is a summing junction, if the output from the secondary winding is in phase with the primary, then

$$I_S = I_{S1} + I_{S2} = \frac{V_D}{R_1} + \frac{nV_D}{R_2}$$

and the 'measured' impedance

$$Z = \frac{V_D}{I_S} = \frac{1}{\frac{1}{R_1} + \frac{n}{R_2}}$$

If the resistors are chosen such that

$$R_2 = nR_1, \text{ then } Z = \frac{R_1}{2}.$$

If the output is antiphase to the primary, then

$$I_S = I_{S1} - I_{S2} \quad \text{and}$$

$$Z = \frac{1}{\frac{1}{R_1} - \frac{n}{R_2}}.$$

Again, if $R_2 = n R_1$, then $Z = \frac{1}{0} = \infty$ (or very high, anyway!).

Either one of these two tests (in-phase or anti-phase) is in itself a sufficient test for verification of turns ratio and phasing, although it is clearly possible to apply switching of the resistors in order to check in both cases.

9.6 ANALOG PROGRAMMING

9.6.1 Introduction

The basic principles of programming have been given in section 9.0 and these should be understood before commencing the detailed analog programming which follows.

The modules used for analog tests will be selected, usually, from the eight described briefly here:

1 Analog Test Module (ATM)

This is the basic analog measurement input module, covering measurement of a.c. volts, d.c. volts, d.c. current, frequency, events, period and pulse width.

2 Power Supply Module (PSM)

This module (in addition to its prime function of powering the UUT) serves as one of three types of analog stimulus. The other two are VIM and SSM.

3 Voltage/Current Source Module (VIM)

In addition to providing a force stimulus, in constant-voltage or constant-current modes, the VIM can be used to sink current from the UUT.

4 Sinewave Source Module (SSM)

This oscillator is programmable in frequency over the range 20Hz to 200kHz and in level from 0 to 8Vrms, with d.c.-coupled and a.c.-coupled outputs. It also provides an unswitched 5V squarewave output.

5 Reed Relay Module (RRM)

This consists of 24 reed relays, (in 12 groups of 2) used for such purposes as signal routing, the connection of dummy loads, and isolation of shunt components.

6 Timer Interface Module (TIM)

This provides single or dual input timing facilities, used for such measurements as delay, phase relationship and voltage slope. Either channel may be used for event counting or for frequency measurement up to approximately 1.5MHz.

7 Impedance Test Module (ZTM)

This provides the in-circuit component test facilities as described in section 9.5.

8 Pulse Generator Module (PGM)

This provides a defined number or a continuous train of pulses, programmable for mark and space times and for level. Its output can also serve as a timing signal for control purposes.

Full specifications of these eight modules are included in Appendix A. Their use is illustrated by means of the example given below.

9.6.2 Example - simple circuit

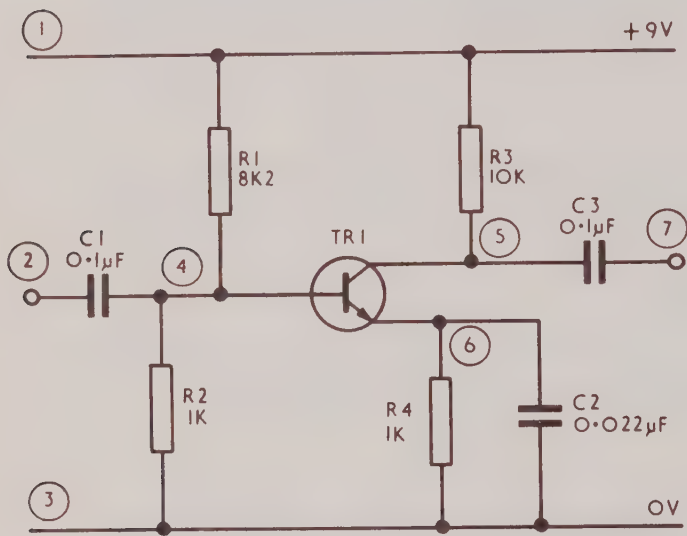


Fig. 9.3 Analog circuit example

Warning

The following contains examples of particular test strategies ie functional GO/NOGO methods with diagnostics by in-circuit testing. The particular test strategy for a UUT will vary, depending on, for example, type of board, primary yield rate, assembly methods, etc. The versatility of the system provides a highly flexible response to the differing test strategies required for different UUTs.

The circuit (Fig.9.3) is to be tested for quiescent conditions with zero input stimulus, its amplification is to be checked with a stimulus input and failure of any test is to implement in-circuit tests to locate the faulty component.

Assuming a maximum sinusoidal output of 6V p-p, =2.12V rms, the required stimulus voltage at each frequency considered should be constant. The circuit has a gain of about 0.5 at 100Hz and 3.75 at 1kHz. For the assumed output of 2.12V at 1kHz, the rms input required would therefore be 565mV. (If a frequency response was

required in some detail, a table of inputs vs frequency would be compiled).

To cater for these tests, the following modules would be needed:

- One Power Supply Module (PSM) for 9V supply
- One Sinewave Source Module (SSM) for 100Hz & 1kHz stimuli
- One Analog Test Module (ATM) for measurement
- One Impedance Test Module (ZTM) for component checks
- One Reed Relay Module (RRM) for switching

Power supply to be applied from (1) to (3)

Sinewave stimulus (2) to (3)

Analog measurements:

- 1 Set PSM, (1) to (3)
- 2 Set SSM, (2) to (3)
- 3 Bias voltage and input level (4) to (3)
- 4 Quiescent collector voltage (5) to (3)
- 5 DC output level (5) to (3)
- 6 AC output (7) to (3)

Component checks:

- C1 (2) to (4)
- C2 Complex Z R4//C2 (6) to (3)
- C3 (5) to (7)
- R1 (1) to (4)
- R2 (4) to (3)
- R3 (1) to (5)
- R4 (6) to (3)
- TR1 e(6), b(4), c(5)

All ZTM pins can remain in contact with UUT as at no time will voltage exceed 15V.
It is good practice to short input via RRM for quiescent check.

Allocate addresses:

PSM	address 1
SSM	address 4 (it will be used with ZTM later)
RRM	address 5
ATM	address 6
ZTM	address 7

Test specification

- 1 Apply 9V power.
- 2 Short input to ground.
- 3 Measure base voltage, reject if not between 0.8 and 1.1V.
- 4 Measure collector voltage, reject if not between 4.7 and 5.3V.
- 5 Remove input short to ground and connect input to sig. generator.
- 6 Set sig. generator to required frequency and level.
- 7 Trim level at input.
- 8 Verify gain.

A simple program would end here.

For a comprehensive program continue from here.

- 9 Deduce new signal frequency and level.
- 10 Check frequency in range, if not go to step 12.
- 11 Return to step 6.
- 12 END sequence.

Reject sequence

- 13 Remove power and stimulus.
- 14 Check all components.
- 15 List out faulty components.
- 16 END sequence.

To implement this test sequence as an ATE program it is very useful to compile a flow chart, as shown in Fig. 9.4.

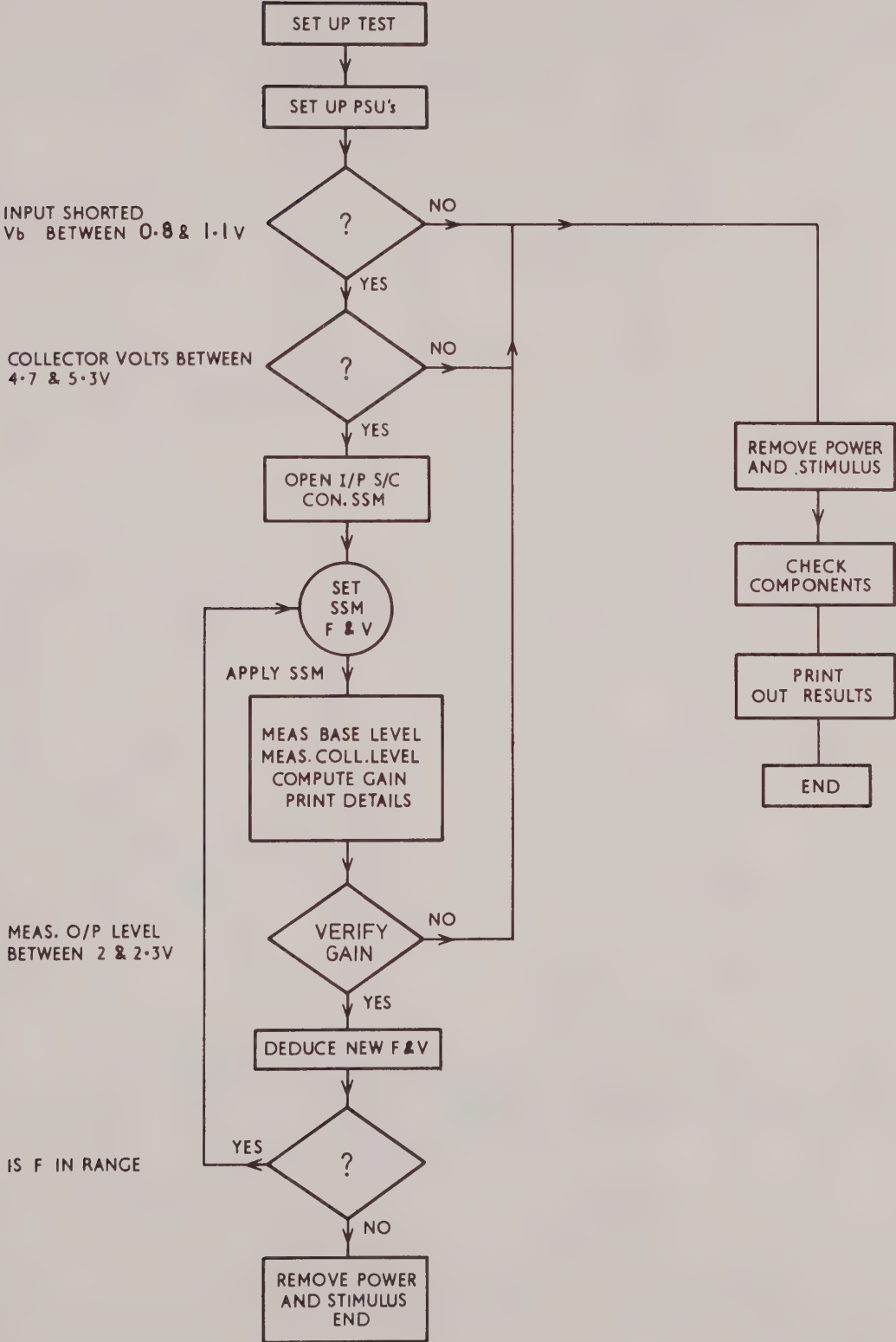


Fig. 9.4 Flow chart example

Test Module Pin allocations:

(1)	PSM+	1/A9	(4)	ATM	6/A2
"	ZTM	7/C2	"	ZTM	7/C1
(1a)	"	7/B2*	(5)	ATM	6/A3
(2)	RRM	5/A2	"	ZTM	7/A1
"	ATM	6/A1	(6)	ATM	6/C1
"	ZTM	7/A5	"	ZTM	7/A2
(3)	PSM-	1/A1	(7)	ATM	6/A4
"	RRM	5/A5**	"	ZTM	7/A3
"	"	5/A3			
"	ZTM	7/C3			
(3a)	"	7/B3*			

RRM 5/A1 - SSM 4/A3
RRM 5/A4 - SSM 4/A1**
ATM 5/A16 - SSM 4/C1 (ATM Ground - see 6.3.1)
PSM 1/B1 to special calibration pin [see section 4.0, 2)]

* Double-pole Guard pin
** Isolation for SSM Ground

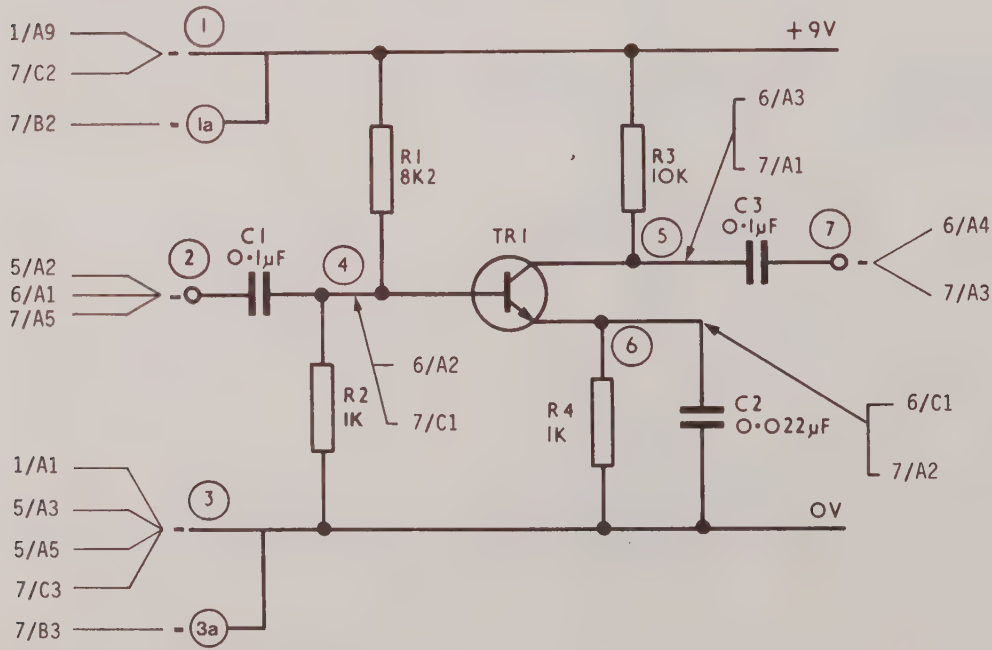


Fig.9.5 Analog circuit example with pin connections

Two test methods are shown. The first, 777777, is a relatively simple test at 5kHz, where the expected stage gain is 7.5, i.e. 280 mV rms stimulus, to 2.12 V rms out. If any functional test performed at line S100, S110, S180 fails, then diagnosis of components begins at S210.

777777

D10 PSM=1

D20 SSM=4

D30 RRM=5

D40 ATM=6

D50 ZTM=7

S60 SU PSU, PS1/A,T09.0V,ILIM100.0MA

S70 SU SINE,MOD4,FREQ5.0KHZ,RMS280.0MV

S75 APP PSU,PS1/A

S80 CL,5/A3: A4

S90 DLY,250.0MS

S100 VER DCV,PIN6/A2,HL1.1V,LL800.0MV

S110 VER DCV,PIN6/A3,HL5.3V,LL4.7V

S120 GOIF, FAIL,S210

S130 OP,5/A3

S140 CL,5/A1

S150 APP SINE,MOD4

S160 DLY,250.0MS

S170 TRIM SINE,MOD4,PIN6/A1,T0280.0MV

S180 VER ACV,PIN6/A4,HL2.18V,LL1.9V

S190 GOIF,NOGO,S210

S200 END

S210 REM ALL

S220 GRD,ALL

S230 VER C,DR7/C1,SN7/A5,HL150.0NF,LL80.0NF,CREF"C1"

S240 VER R,DR7/C2,SN7/C1,HL8.5KOHM,LL7.9KOHM,CREF"R1"

S250 VER R,DR7/C3,SN7/C1,HL1.1KOHM,LL900.0OHM,CREF"R2"

S260 VER NPN,E7/A2,B7/C1,C7/A1,DLY10.0MS,CREF"TR1"

S270 VER R,DR7/C2,SN7/A1,HL10.5KOHM,LL9.5KOHM,CREF"R3"

S280 VER C,DR7/A1,SN7/A3,HL150.0NF,LL80.0NF,CREF"C3"

S290 VER R,DR7/C3,SN7/A2,HL1.02KOHM,LL998.0OHM,DLY100.0MS,CREF"R4"

S300 VER Z,FREQ200.0HZ,DR7/C3,SN7/A2,HL980.0OHM,LL975.0OHM,CREF"R4/C2"

S310 END

By comparison, program 666666 checks gain at all frequencies (100Hz, 500Hz, 1kHz, 5kHz, 10kHz, 50kHz and 100kHz) against stage gain expectancies. Two string variable stores (N4 and N3) have been used to contain strings relating to gain (N4) and frequency (N3) from which the relevant numerical data is retrieved.

```
i.e.      S80 LET, M1=0.0
          S90 LET, M2=M1+3.0
          "
          "
          "
          S220 SLET, M5=N4(M1:M2)  allocate the first 4 characters from
                                   N4 (00.5) to M5, which represents them
                                   as 500.0M., the expected gain (in S60)
                                   for the corresponding frequency (in S70).

          S100 LET, M3=0.0
          S110 LET, M4=M3+5.0
          "
          "
          "
          S230 SLET, M6=N3(M3:M4)  N3(0:5) =0100.0
                                   in M6 becomes 100.0 , Hz
```

666666

D10 PSM=1

D20 SSM=4

D30 RRM=5

D40 ATM=6

D50 ZTM=7

S60 SLET, N4="00.5,02.5,03.7,07.5,12.5,37.5,62.5"

S70 SLET, N3="0100.0,0500.0,001.0K,005.0K,010.0K,050.0K,100.0K"

S80 LET, M1=0.0

S90 LET, M2=M1+3.0

S100 LET, M3=0.0

S110 LET, M4=M3+5.0

S120 SU PSU,PS1/A,T09.0V,ILIM100.0MA

S130 CL,5/A3

S140 APP PSU,PS1/A

S150 DLY,250.0MS

S160 VER DCV,PIN6/A2,HL1.1V,LL800.0MV

```
S170 GOIF,NOGO,S440
S180 VER DCV,PIN6/A3,HL5.3V,LL4.7V
S190 GOIF,NOGO,S440
S200 OP,5/A3
S210 CL,5/A1
S220 SLET, M5=N4(M1:M2)
S230 SLET, M6=N3(M3:M4)
S240 SU SINE,MOD4,FREQ M6HZ,RMS280.0MV
S250 APP SINE, MOD4
S260 DLY,250.0MS
S270 TRIM SINE,MOD4,PIN6/A1,T0280.0MV
S280 MEAS ACV,PIN6/A4,VR5,INTO M11
S290 LET, M11=M11/280.0M
S300 LET, M8=M5*900.0M
S310 LET, M7=M5*1.1
S320 COMPFAIL, M11,HL M7,LL M8
S330 GOIF,NOGO,S440
S340 COMP, M2,LL32.5
S350 GOIF,GO,S410
S360 LET, M1=M1+5.0
S370 EX,S90
S380 LET, M3=M3+7.0
S390 EX,S110
S400 GOTO,S220
S410 PRINT,W10,"FUNCTIONAL"
S420 PRINT,W10,"TEST"
S430 END
S440 REM ALL
S450 DLY,250.0MS
S460 PRINT,W10,"FAIL"
S470 EX,S410
S480 EX,S420
S490 GRD,ALL
S500 OP,5/*
S510 UNFLAG, F99
S520 VER C,DR7/C1,SN7/A5,HL150.0NF,LL80.0NF,CREF"C1"
S530 VER R,DR7/C2,SN7/C1,HL8.5KOHM,LL7.9KOHM,CREF"R1"
S540 VER R,DR7/C3,SN7/C1,HL1.1KOHM,LL900.0OHM,CREF"R2"
```

```
S560 LET, M14=580.0
S570 GOIF, F99,S780
S580 UNFLAG, F99
S590 VER NPN,E7/A2,B7/C1,C7/A1,DLY10.0MS,CREF"TR1"
S600 VER R,DR7/C2,SN7/A1,HL10.5KOHM,LL9.5KOHM,CREF"R3"
S610 VER C,DR7/A1,SN7/A3,HL150.0NF,LL80.0NF,CREF"C3"
S620 LET, M14=640.0
S630 GOIF, F99,S780
S640 EX,S580
S650 LET, M19=2.0*200.0*22.0N*(355.0/113.0)
S660 LET, M18=1.0M+M19
S670 LET, M17=1.0/M18
S680 LET, M16=M17*1.1
S690 LET, M15=M17*900.0M
S700 VER R,DR7/C3,SN7/A2,HL1.02KOHM,LL998.0OHM,DLY100.0MS,CREF"R4"
S710 VER Z,FREQ200.0HZ.DR7/C3,SN7/A2,HL M160HM,LL M150HM,CREF"R4/C2"
S720 GOIF, F99,S770
S730 GOIF, F7,S760
S740 PRINT,W10,"COMPONENTS"
S750 PRINT,W10,"TESTED OK!"
S760 END
S770 FLAG, F7,IF F99
S780 PRINT,W10,"COMPONENT"
S790 PRINT,W10,"FAILURE!"
S800 GOTO,S M14
```

Using this technique, a routine can be evolved from S220 to S400 such that the predicted gain can be checked at all listed frequencies. Any failure will invoke the Diagnostic route starting at S440, and splitting the components into groups (S520 - 540, S590 - 610) and calculating the impedance (Z) at 100 Hz of the combination R4 and C2 prior to testing it.

The two methods illustrate a simple and a sophisticated approach, respectively.

9.6.3 Continuity using ZTM's

A test program developed to functionally stimulate and check a circuit cannot, on the A8000, ascertain and correctly report malfunctions caused by track DISCONTINUITY (open circuits) or unacceptable nodal CONTINUITY (short circuits). It tends to report them as functional failures, or as component failures. Connectivity failures can only be logged by the use of a second program requiring a different test rack population.

When building a program on A8315, a connectivity chart is generated at the appropriate time by selecting "CONTINUITY" from the IPG. At this time, requests will be made for the threshold value (between 3 and 5K ohms) below which a short is assumed, and above which the assumption is open circuit, and also the maximum node to node capacitance. Default values are 10 ohms and zero. The input of a capacitance value allows a delay to be built in to take up capacitor charge/discharge times. The program line will, at this stage, be similar to:

```
S200    CONTINUITY, LRN, TL 9.0 OHMS, MAX C 1000 UF
```

This indicates that the line has been generated, but the connectivity table has not yet been learned. This will occur later. When the program has been generated, a known good unit must be offered to the test bed. This is obviously vital in order to verify and de-bug the functional program, but the good unit will be needed to build the connectivity table.

When the program is run and a CONTINUITY line is encountered, several things occur:

- FIRST: All stimulus modules are initialised, that is re-set to their prime state and removed from connection to the UUT. This does not occur for RRM's which continue in their current state.
- SECOND: The equipment looks for the field LRN in the CONTINUITY statement, and acts accordingly.
- THIRD: On exit from the CONTINUITY statement all stimulus modules remain in their initialised and unconnected state, except RRM's which remain as they were prior to the CONTINUITY statement occurring.

When the system inspects the CONTINUITY statement for LRN and finds it, no tables have been generated. The system will create a table by checking the pin-to-pin connectivity for ALL ZTM's addressed by the test jig even if declared as ITM's, and remove the LRN field from the statement.

If, on inspection of the CONTINUITY statement, the system does not find LRN, then tables have been produced (i.e. the program has been run before to learn the table data) and all addressed ZTMs are processed on a pin-to-pin basis to verify compliance with the table. Any discrepancy will be reported.

One further action occurs when leaving the CONTINUITY statement. All pins on declared ZTM's are returned to tri-state. If the ZTM's were in fact declared as ITM's then the pins are initialised as guards.

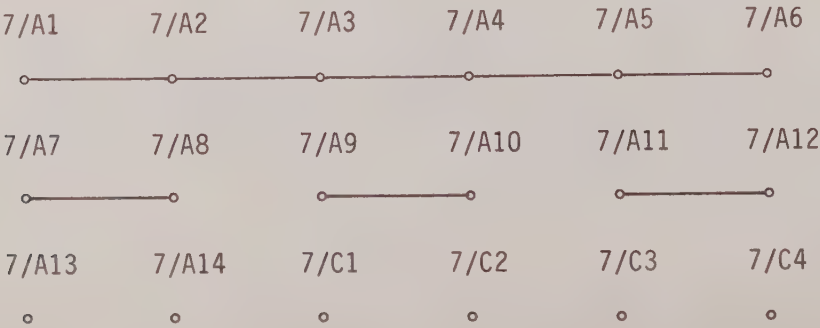
A program print out, or a list, will show the CONTINUITY statement in one of two forms:

- A) S200 CONTINUITY, LRN, TL 3.0 OHMS, MAXC 100 UF, or
- B) S200 CONTINUITY, TL 3.0 OHMS, MAXC 100 UF

the second (B) indicating that learning has taken place, and that tables are complete. To gain visual access to the tables, the typing of the word ALL after the list, print or output request (L, P, O,) will insert as a subset of the CONTINUITY statement, the connectivity table. It differs from the SOM table inasmuch as it lists only CONTINUITY - not DISCONTINUITY - and it also registers CONTINUITY to ground.

Because the learning technique differs from that for SOM's, the error reporting technique is also different, and may, at first sight, seem to require more analysis. This is misleading, as the reporting technique is, in fact, more representative of the fault, and can be used to indicate more precisely the position of the fault. It can also detect shorts to ground.

An example:



ZTM program

```

D10 ZTM= 7
S20 CONTINUITY, TL 10 OHMS
  Q1  (20) S/C List
  Q10 (20) 7/A1:7/A2:7/A3:7/A4:7/A5:7/A6
  Q20 (20) 7/A7:7/A8
  Q30 (20) 7/A9:7/A10
  Q40 (20) 7/A11:7/A12
S30      Next Statement

```

The ZTM program is considerably more efficient than an SOM program in terms of storage space, but the error reporting is different. Assume track from 7/A3 to 7/A4 is broken, the report will be:

```
TP ABC123
```

```
*****
```

```
FAIL 0020
```

```
OPEN CIRCUIT PAIRS
```

```
7/A1 - 7/A4
```

```
7/A1 - 7/A5
```

```
7/A1 - 7/A6
```

```
SHORT CIRCUIT PAIRS
```

```
7/A4 - 7/A5
```

```
7/A4 - 7/A6
```

For full
explanation
see page 9-31.

Shorts to ground may not be picked up by SOM but, for ZTM, current flowing from a drive pin must go somewhere. If it flows into a sense pin, then a short drive-to-sense is assumed. If it does not flow into a sense pin, then the system will assume a short to ground. Should this be detected at the learning stage, then it becomes part of the table but, if detected during program run stage and is not a part of the table, it is reported.

With COMBAT ZTM CONTINUITY generator, some editing is possible. Editing of the tables is not allowed, and access is prohibited, but editing of the CONTINUITY statement is possible, and is as follows:

```

Key in S 200 carriage return,

S200 CONTINUITY, TL 9.0 OHMS, CVAL 100 UF

S200 □    (Cursor)

```

The character 'X' for eXterminate may be used at any point. If invoked before the end of the key word (CONTINUITY), the whole line and table may be cancelled, and the equipment will request the user to

DELETE LEARN CONTINUITY PROGRAM (Y or N)?

Typing Y deletes the statement and table, and N returns to FUNCTION ? level. However, the continuity parameters may be edited without the pattern having to be relearned. As an example, consider the statement

S140 CONTINUITY, TL100.00HMS,MAXC25.0UF

Editing the part after "CONTINUITY," is achieved as described in section 8.2 (Functional Line Editor). In this example, if 100.00HMS is to be changed to 325.00HMS, the cursor would be moved to the '1' of 100 and an input keyed in of

3C325 <CRET>

The threshold field will then read TL325.00HMS, .

It must be remembered that stimulus levels are determined by programmed threshold values. The level is 1mV for each ohm of threshold, with a resolution of 2.5mV. For example:

TL3.0 OHMS	:	2.5mV
5 OHMS	:	5mV
8 OHMS	:	7.5mV
5K OHMS	:	5V

Consequently, significant changes in threshold, subsequent to the Learn process, may cause a program to fail.

Occasionally, apparently contradictory fault reporting can arise, such as:

OPEN CIRCUIT PAIRS

6/C1 - 5/C3

SHORT CIRCUIT PAIRS

6/C1 - 5/C3

This will usually be overcome by editing a CONTINUITY statement to use a different MAXC value.

Note: Whenever an RRM is declared, statements must be entered to establish 'close' and 'open' for all contacts used (see 9.9.7).

9.7 DIGITAL PROGRAMMING

9.7.1 Introduction

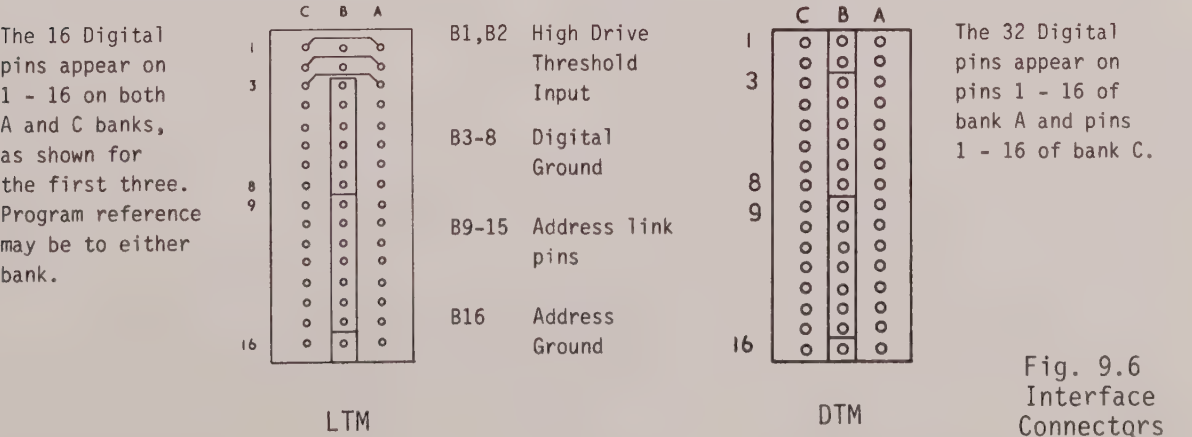
Digital circuits are tested using a functional test program. Under control of such a program, the system will power up the unit under test, activate the circuit inputs and check the outputs to ensure that the circuit operates correctly. The set of input conditions (stimulus) and corresponding expected good output states (response) constitute the main body of the program and, together with any required diagnostics, form the complete test procedure. Any LTM or ~~DMT~~ pins remaining connected to ZTM or ITM pins will make in-circuit measurements inaccurate. Consequently, digital test pins should be disconnected, using RRM, when in-circuit testing is being done.

9.7.2 Fundamentals

Power to the UUT is from programmable supplies under control of the Power Supply Module(s) (PSM). Logic levels on the inputs and outputs are driven and sensed respectively using Logic Test Modules (LTMs) or Digital Test Modules (DTMs).

Each LTM has a 16-pin capability and each DTM 32 pins, where each pin can be either a driver or sensor, and is programmable over a range which allows the system to test digital circuits operating in the voltage range 0 to +15V.

With currently available modules, the high, or logic 1, drive level is determined by wiring in the jig such that the required logic 1 voltage level (e.g. 5V for TTL) is taken from the appropriate PSM providing power to the circuit, to a reserved pin (B1 or B2) on the test module (see Fig. 9.6). By default, the LTMs when used in the system assume a 3.3V level; when using DTMs, however, the reference voltage pin B1 must always be wired to the power supply for the appropriate logic family.



OV from the PSM is similarly connected to any of the digital ground pins on the LTM or DTM interface connector. (See section 6.3.1).

LTM pins A1 to A16 are internally connected to pins C1 to C16 respectively and are the 16 digital test pins: the program can refer to either the A or C pin bank. DTM pins A1 - A16 and C1 - C16 must be individually specified.

9.7.3 Pin Referencing

As described, each module has a defined number of test pins and as many modules as required for a particular circuit are used and defined as usual in the declaratives at the beginning of the program.

Having declared the modules and their addresses, there are two ways in which the pins can be referenced in the test program.

- 1) The first is the normal method and of the general form

x/by (e.g. 5/A12)

where: x is the module address as per the declarative (and the jig-wired address for the specific module)

b is the bank letter - and will be A or C

y is the pin number in the range 1-16.

- 2) The second method of specifying the pins in the program is to assign a name to a pin or group of pins prior to the start of the stimulus/response section of the program and then subsequently to refer only to the defined name.

The general form of each executable statement to name a pin or group of pins is:

DEF NAME, #a(t,b) = pinlist

where: a is the selected name to a maximum of 6 ASCII characters (e.g. CLOCK, RESET, DBUS etc.)

t is the no of bits - i.e. the number of pins in the pinlist

b is the number base for any data which will be applied or sensed on the named list and can be

B - binary

O - octal (letter O, not zero)

D - decimal
default - hexadecimal

The number base (and its preceding comma) is optional and if it is not specified, the system will use hexadecimal.

Pinlist - a list of the pins in the normal form as previously described.

Since these statements (one DEF NAME per line) are executable, each must be preceded by S and a line number.

A total of 64 such names is available.

EXAMPLES

S50 DEF NAME, #CLOCK(1) = 9/A4

S100 DEF NAME, #SELECT(4,B) = 8/A16:15:14:13

S63 DEF NAME, #DBUS(8,D) = 9/A2:A3:A4:A5, 8/A16:15:14, 9/A1

Where a pinlist consists of more than one pin, the leftmost pin in the list is the most significant bit with regard to any pattern (to the base defined) which may be applied or sensed.

9.7.4 Basic Programming

Referring to the circuit of Fig.9.7 and its operation represented by the truth table, the program to test this simple circuit will be a series of test statements which will step the circuit through the truth table and check that the logic state of the output is correct as defined by the truth table.

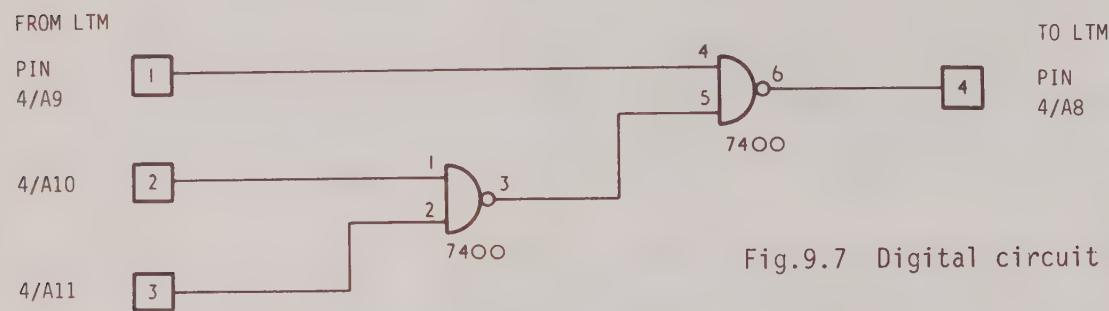


Fig.9.7 Digital circuit example

	INPUTS			OUTPUT
	1	2	3	4
PATTERN 1	0	0	0	1
PATTERN 2	1	0	0	0
PATTERN 3	1	1	0	0
PATTERN 4	1	1	1	1
PATTERN 5	1	0	1	0

Note that to ensure that all possible faults are propagated to the output (and therefore detectable by the system), the programmer must define the input patterns and the expected state of the output at each step. Also note that it may not be necessary to provide all possible combinations of input data to achieve this. In fact, with circuits of only moderate complexity, it is usually the case that the majority of possible input data patterns will not propagate any fault-related data to an output.

Ignoring the preliminary set-up statements for the sake of clarity, the test program statements to implement the truth table would be:

For Pattern 1

```
S80 SET LO 4/A9:A10:A11, HI 4/A8,VER.
```

This will set a logic LO on all three input connections and instruct the machine to expect a logic HI on the output.

VER is the instruction to sample the state of the pins and if the UUT outputs (or inputs for that matter) are not as programmed, the tester will indicate a faulty UUT.

For Pattern 2

S90 SET HI 4/A9, LO 4/A8, VER

This statement will change input pin 1 of the UUT from a logic LO to a logic HI and instruct the system to expect that the output (pin 4 of the UUT) should go to a logic LO. Again the VER will actually test that this condition exists.

Note that it is not necessary to re-specify the states of pins which do not change in this step - their condition will be maintained from the previously defined state.

For Pattern 3

S100 SET HI 4/A10, VER

This will drive input pin 2 HI - there should be no change on output pin 4 and the VER will check this. Should the output not remain LO as previously specified in pattern 2, this will indicate a faulty UUT.

For Pattern 4

S110 SET HI 4/A11:A8, VER

This will drive input pin 3 HI and since input pins 1 and 2 are still both HI, will result in output pin 4 going HI. Again the VER tests this to be so.

For Pattern 5

S120 SET LO 4/A10:A8 VER

This will change input pin 2 from a logic HI to a logic LO and accordingly change the output pin 4 to a logic LO. Again the VER makes the check.

Prior to these statements will be the declaratives and set-up statements, so the complete program will be:-

D 10	PSM = 1/1
D 20	LTM = 4/3
S 30	SU PSU PS1/A TO 5.0V
S 40	SU SENSE MOD4 TO 1.8V
S 50	APP PSU PS1/A
S 60	DLY 250.0MS

9-54

```

S 70      SET DR 4/A9:A10:A11, SN 4/A8
S 80      SET LO 4/A9:A10:A11, HI 4/A8 VER
S 90      SET HI 4/A9, LO 4/A8, VER
S100      SET HI 4/A10, VER
S110      SET HI 4/A11:A8, VER
S120      SET LO 4/A10:A8, VER
S130      END

```

Statement S40 is the command to program the threshold voltage to determine the logic HI and logic LO levels to be used by the module pins and, as shown, specifies a single level. If no threshold level has been specified in the program, the system automatically assumes 1.5V as a single threshold level.



It is also possible to program dual thresholds. For example, S40 could have been:

```
SU SENSE MOD4 HL 2.2V LL 0.8V
```

in which case any pin sensed as lying between the two specified thresholds is considered indeterminate and therefore faulty.



In both examples of the threshold setting, the term MOD4 refers to the module address.

Statement S70 defines which pins are drivers (i.e. providing logic input signals to the UUT) and which are sensors.

If symbolic programming is used, the names would have to be defined and possible examples could be:-

```
S 25    DEF NAME#INPUTS (3)  = 4/A9:A10:A11
S 26    DEF NAME#OUTPUT (1) = 4/A8
```

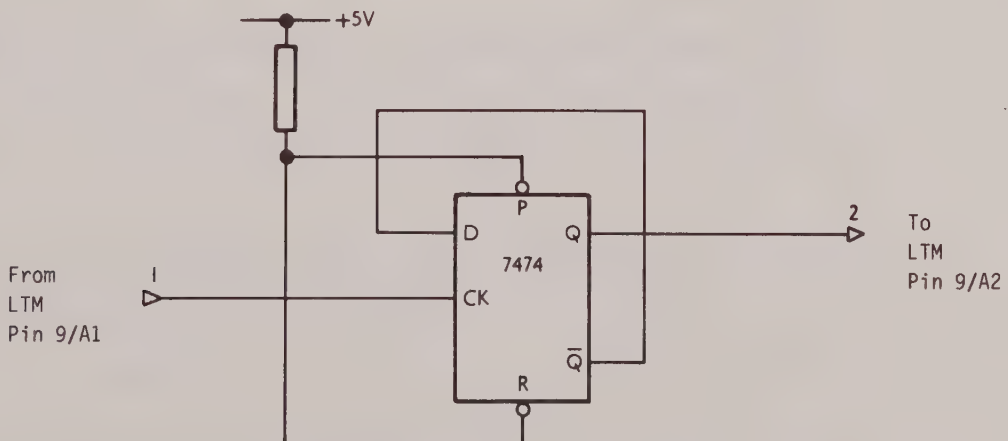
Then the test steps would be:-

```
S 70    SET#INPUTS  = DR,#OUTPUT = SN
S 80    SET#INPUTS  = 0,#OUTPUT  = 1, VER
S 90    SET#INPUTS  = 4,#OUTPUT  = 0, VER
S100    SET#INPUTS  = 6, VER
S110    SET#INPUTS  = 7,#OUTPUT  = 1, VER
S120    SET#INPUTS  = 5,#OUTPUT  = 0, VER
```

9.7.5 Initialisation

To ensure reliable and repeatable testing, it is absolutely imperative that the circuit being tested is set to a known state (or initialised) prior to performing any tests which set the pass/fail highway (i.e. executing a VER).

In the simple example used so far, this has not been a problem since it does not have any sequential logic. However, even the simple example below will require a specific initialisation routine since, depending on device characteristics, it could power up with the flip flop set in either state.



The circuit is a simple divide by 2 but the set and reset inputs to the flip flop are tied up. If either of them were available, initialisation would be very easy.

However, in the configuration shown, it is necessary to test the output and, if it is not in the required state, to clock the device before proceeding with the rest of the test program.

Ignoring the set-up instructions, a possible initialisation routine to set the flip flop to a known state could be:-

```

S100  SET  DR 9/A1, SN 9/A2
S110  SET  LO 9/A1:A2, MEAS
S120  GOIF GO S140
S130  SET  CLK 9/A1
S140
      ↓
    REST OF PROGRAM

```

The sequence of operation is therefore:-

S100 - define inputs and output.

S110 - define initial state of input and required state of output and test using the measure (MEAS) statement. This is similar to VER as already used but sets only the transitory flags (GO/NOGO etc.). It will not cause a failure if the output is not in the required state since, particularly in this case, the wrong state at this stage does not necessarily indicate a fault.

S120 - a conditional jump which will be executed if all the previously specified conditions are met, i.e. the output is a logic L0 (and incidentally the input must also be as previously specified - i.e. logic L0).

If the conditions are met the circuit is in the required (initialised) state and the next statement to be executed will be S140. If the output is not a logic L0 the jump at S120 will not be executed. Instead S130 will be performed.

S130 - issues one clock pulse on the input and will change the state of the flip flop (assuming no faults).

Since this statement will only be executed if the flip flop was not initialised as required, one clock pulse must put the flip flop in the required state (again - assuming no faults) so the test program can continue.

Any faults or non-operation of the flip flop would be detected by a VER either in S410 or later in the test program.

Trouble associated with digital test programs intermittently failing good boards can often be traced to inadequate initialisation. The source of trouble can be something as simple as an uninitialised cross-coupled latch or reading from a location in RAM which has not previously been written to by the Test Program. It is also good practice to test that all drive pins can be successfully driven high and low before proceeding with testing. This is best done after power on, prior to initialisation. Remembering that the start state of all digital pins is sense, neglected, low, there is an easy way to do this.

Example:

```

SU      PSU .....
APP     PSU .....
DLY     .....      Minimum 300 msec
DEF NAME #LINE(4)=.....
S100 SET, #LINE=DR, DR8/A4,8/A6.....VER Sets drive pins and monitors them
                                           and checks they drive low OK.
      SET,HIAL,XALL                      Sets all pins high and monitored,
                                           ignores all pins.
      EX S100                            Respecifies drive pins which
                                           remonitors them, and tests they
                                           can drive high OK.

      Initialisation

```

9.7.6 Program loops & algorithms

By use of symbolic names and the floating-point numerical stores (M0 - M19), program loops can easily be developed to cut down the number of test statements needed. The simple example below would generate a walking one pattern on eight inputs called #DBUS.

```

S50  DEF  NAME  #DBUS(8)= 6/A1:A2:A3:A4:A5:A6:A7:A8
S60  SET   #DBUS = DR
S70  LET   M1 = 1
S80  SET   #DBUS = M1
"    }
"    }   TESTS AS REQUIRED
"    }
"    }
S150 LET   M1 = M1 * 2

```

```

S160  COMP  M1  HL  129
S170  GOIF  GO  S80
"
"  }      REST OF TEST PROGRAM
"

```

The use of pattern decrement and increment instructions (PD & PI) provide more flexibility.

Example of algorithmic testing

Circuit

8 bit data latch, with low latch enable (on pin 8/A15).

```

Initialisation      100 DEF NAME#DATAIN(8) = 8/A1:A2:A3:A4:A5:A6:A7:A8
                   120 DEF NAME#DATOUT(8) = 9/A1:A2:A3:A4:A5:A6:A7:A8
                   130 SET,DR8/A15,HI8/A15,
                     SET #DATAIN = DR

Walking
'1'                140 LET M13 = 1
                   150 SET,#DATAIN = M13,CLK8/A15,#DATOUT = M13,VER
                   160 LET M13 = M13 * 2
                   170 COMP M13,HL 129
                   180 GOIF GO S150

Walking
'0'                190 LET M14 = 1
                   200 LET M13 = 255 - M14
                   210 EX S150
                   220 LET M14 = M14 * 2
                   230 COMP,M14,HL 129
                   240 GOIF GO S200
                   END

```

9.7.7 Mixed family logic

The ATE will test PCBs containing mixed family logic (e.g. TTL & CMOS) but the programmer must note that the logic HI levels and threshold settings used apply to all the pins of the module to which they relate. Thus to test a board containing say 5V TTL & 12V CMOS it will be necessary to wire TTL connections to modules programmed to TTL levels and the CMOS connections to other modules programmed for CMOS levels.

9.7.8 Time measurements

The standard system has, inherently, the ability to make timing measurements using the in-built two channel counter-timer.

By use of the CONnect statement, any LTM pin can be connected to either the

A or B inputs of the timer, thus allowing measurements of time periods on one pin (e.g. start on A - stop on A) or between two pins (start on A - stop on B), or frequency/event measurements. For further details, refer to sections 9.9.14 and 9.9.15.

9.7.9 Hybrid PCB testing

PCBs which contain digital and analog circuitry can be tested on the system since it is possible to stimulate digitally and measure analog responses (or vice versa) by using a mixture of the required modules in the test rack.

There is no problem or penalty associated with mixing digital and analog test statements, but it must be noted that the automatic digital diagnostic (ADD) facility will not be invoked by detection of a purely analog failure.

Test points at the analog/digital boundaries of hybrid PCBs will obviously greatly aid program generation and also provide better diagnostic possibilities.

Example of hybrid testing

To test an 8 bit DAC (0-5V) to $\pm\frac{1}{2}$ 1sb. Output connected to ATM pin 6/C9.

```

110 DEF NAME #DACIN(8) = 10/A14:A15:A16:29/A1:A3:A5:A7:A9
120 LET M16 = 5/255                                calculates resolution of dac
130 LET M1 = 1
140 SET, #DACIN = M1,VER
150 MEAS DCV, PIN 6/C9 INTO M12
160 LET M15 = M16 * M1                                calculates nominal o/p
170 LET M17 = M15 + M16/2                            high limit
180 LET M18 = M15 - M16/2                            low limit
190 COMPFAIL, M12, HL M17, LL M18
200 LET M1 = M1 * 2
210 COMP, M1 HL 129
220 GOIF GO S140
230 END

```

Example shows only check of walking '1'. Full pattern check could be performed by changing S200 to

```
LET M1 = M1 + 1
```

and by changing S210 to

```
COMP, M1, HL 255.5
```

In this case, a monotonicity test could also be performed.

NOTE:

The high limit shown above is deliberately 255.5, to avoid rounding errors in progressive incremental loops using algorithmically generated numbers. In this example, 1 is added to the store before testing it. The last number to permit reiteration round the loop is 255, and therefore the absolute high limit is also 255. Since this is progressively calculated from adding 1 two hundred and fifty five times, the final may be 255 plus a small fraction in the store. A test against a high limit of exactly 255 may fail. To avoid this, limits should be set to $\pm \frac{1}{2}$ of the actual limit.

In the other examples in this section involving the walking 1 algorithm, the limit has been 129. This is because the sequence is multiply by 2, e.g. 64, 128, 256. Equally the limit could have been 128.5, or any integer 129-255 inclusive. The fact that rounding errors result in non-exact integers has no effect on the translation to digital patterns, as the system automatically rounds to the nearest integer.

9.7.10 Program statement timing/operation

The pin commands basically operate on software tables held in RAM. Periodically the information is transferred from these tables to the pin hardware. Responses are tested by comparing actual pin state with the content of the tables. This 'sample' is caused by either the VER or MEAS commands.

1	APP	actions the hardware			
2	MEAS	"	"	"	performs a sample & sets GO/NOGO flags
3	VER	"	"	"	performs a sample & sets GO/NOGO flags & PASS/FAIL flags
4	CLK	"	"	"	prior to clock burst
5	"READ"	"	"	"	and typically indicates SET store = #name
6	end of statement actions the hardware unless immediately preceded by any of 1, 2 or 3				

When using ADD, the signatures are updated (i.e. a sample is taken, assuming SIGON) at every step of 1 to 6 above.

Pins of LTM and DTM are APPLIED in groups as shown below, the module sequence being the declared order (NOT by linked address, nor by position in test rack). For the first declared LTM/DTM module, the first group of pins are applied without a deliberate delay (i.e. within skew etc). Subsequently, the maximum

times (i.e. worst case), with pin changes in each group, are as follows:

LTM

Pins A & C 1-4	:	430 μ s after previous modules last pin
" " 5-8	:	10 μ s later
" " 9-12	:	310 μ s later
" " 13-16	:	10 μ s later

DTM

Pins A1 - A8	:	153 μ s after previous modules last pin
" A9 - A16	:	31 μ s later
" C1 - C8	:	31 μ s later
" C9 - C16	:	31 μ s later

9.7.11 Decimal-Hex. conversion

Introduction

Users will realise that with such statements as:

Set #DATA = M1

M1 can contain only decimal data. In certain circumstances it is more convenient to be able to use hex and the program (as per 000002 which follows) will enable this for 8 bit hex to decimal conversions.

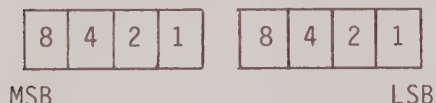
Solution

The program uses string store NØ to hold the hex data as pairs of hex characters (up to a max. of 39 pairs). The data must be terminated with ; (or any other non-hex character you may wish to put in line S60).

The general form of the data is:

XX YY ZZ ;

where each pair (e.g. XX) is of the form:



In sample program 000002, S10 invites the user to enter a data string but, in a user program, this could obviously be a statement such as:

S10 SLET NØ = 12FF378C2D9EB30004;

NOTE: Zeroes must be included as required to pad out the numbers to 8 bits.

S20 to S50 initialise the memory ref. variables used.

M1 is the character string pointer.

M2 is temporary storage for the decimal equivalent being calculated.

M3 is a flag (and multiplier) to indicate which 4 bits are being handled at any particular time. It is set initially to 16.

M4 contains the decimal equivalent of each hex byte at the end of each time through the loop.

S60 checks to see if the end of the string has been reached - if so S70 jumps on to S440 which checks to see if the high or low order nibble is under examination. (A nibble is half a byte - i.e. 4 bits).

If S450 does not G0 it indicates that only 1 character of the required pair was decoded (i.e. the ; was used when another hex digit was expected).

If S450 does G0 everything has terminated correctly.

If S70 does not G0, lines S80 through S190 check for A to F (hex) and G0 accordingly to S230 - S330, to compute the decimal equivalent (in M2) and then to S340.

S340 checks to see if the 1st or 2nd nibble is being actioned. If it is the first nibble it will not G0 at S350 so

S360 stores 1st nibble decimal equivalent (M2) in M4

S370 sets the nibble flag (and multiplier)

S380 increments the string pointer, and

S390 causes a re-run from S60 to get the 2nd nibble of the current byte.

If lines S80 through S190 do not detect A to F, then S200 checks that the character is in the range 0-9 by outputting INVALID NUMBER if it is outside the 0-9 range.

S210 computes the decimal value of the 0-9 character (in M2) by using M3 as a multiplier (16 if higher order nibble- 1 if low order nibble).

S220 causes a jump to S340 where again a check is made as to which nibble is being computed. If it is the 2nd nibble of a byte, S350 will GO to S400 where the decimal value of the first nibble is added and at S410 increment the string pointer.

S420 in the listing merely prints the value of M4 at this point but it is here that the operator would use the SET #DATA type statement, clock it, verify it etc., but then return to S30 (as S430 does) to get the next byte in the string.

This is only one method of using the current A8000 software to achieve the desired result - the code can be modified as required.

Simple changes will allow the use of 4 bits or 12 or 16 bits etc.

Error Reporting

The program can return 2 applications errors at run time. They are:

1 INVALID NUMBER which indicates a non-hex digit in the input string

or

2 STRING INDEX ERROR which indicates no terminator (i.e. missing ;).

Sample program

```

000002
S10 IN,P1, N0
S20 LET, M1=0.0
S30 LET, M2=0.0
S40 LET, M3=16.0
S50 LET, M4=0.0
S60 SCOMP, N0(M1),";"
S70 GOIF,G0,S440
S80 SCOMP, N0(M1),"A"
S90 GOIF,G0,S230
S100 SCOMP, N0(M1),"B"
S110 GOIF,G0,S250
S120 SCOMP, N0(M1),"C"
S130 GOIF,G0,S270
S140 SCOMP, N0(M1),"D"
S150 GOIF,G0,S290
S160 SCOMP, N0(M1),"E"
S170 GOIF,G0,S310
S180 SCOMP, N0(M1),"F"
S190 GOIF,G0,S330
S200 SLET,M2= N0(M1)
S210 LET, M2=M2* M3
S220 GOTO,S340
S230 LET, M2=M3*10.0
S240 GOTO,S340
S250 LET, M2=M3*11.0
S260 GOTO,S340
S270 LET, M2=M3*12.0
S280 GOTO,S340
S290 LET, M2= M3*13.0
S300 GOTO,S340
S310 LET, M2= M3*14.0
S320 GOTO,S340
S330 LET, M2= M3*15.0
S340 COMP, M3,HL1.0,LL1.0
S350 GOIF,G0,S400
S360 LET, M4= M2
S370 LET, M3=1.0
S380 LET, M1= M1+1.0
S390 GOTO,S60
S400 LET, M4= M2+ M4
S410 LET, M1= M1+1.0
S420 OUT,P1,"WHICH IS", M4," IN M4"
S430 GOTO,S30
S440 COMP, M3,HL16.0,LL16.0
S450 GOIF,G0,S490
S460 PRINT,W40,"NOT A HEX CHAR PAIR"
S470 OUT,P1,"NOT A HEX CHAR PAIR"
S480 END
S490 OUT,P1,"CONTINUE"

```

9.8 IEEE-488 CONTROL FACILITIES

The system provides full IEEE-488 control facilities and, in conjunction with the string store handling capabilities, is able to control all types of IEEE-488 compatible instruments. However, the individual instrument interface specifications must be studied in order to ensure complete compatibility and safety. This is of particular importance in connection with Device Clear arrangements [see section 9.8.1, IEEE CMD, 1).].

The IEEE controller card interfaces the ATE to the IEEE-488 bus. The card conforms to the IEEE 488-1978 standard, and offers the following subsets.

SH1	-	SOURCE HANDSHAKE	-	COMPLETE CAPABILITY
AH1	-	ACCEPTOR HANDSHAKE	-	COMPLETE CAPABILITY
T4	-	TALKER FUNCTION	-	BASIC TALKER
TEØ	-	EXTENDED TALKER	-	NO CAPABILITY
L2	-	LISTENER FUNCTION	-	BASIC LISTENER
LEØ	-	EXTENDED LISTENER	-	NO CAPABILITY
SRØ	-	SERVICE REQUEST FUNCTION	-	NO CAPABILITY
RLØ	-	REMOTE LOCAL FUNCTION	-	NO CAPABILITY
PPØ	-	PARALLEL POLL FUNCTION	-	NO CAPABILITY
DCØ	-	DEVICE CLEAR FUNCTION	-	NO CAPABILITY
DTØ	-	DEVICE TRIGGER FUNCTION	-	NO CAPABILITY
C1,C2,C3,C4,C28	-	CONTROLLER FUNCTION	-	SYSTEM CONTROLLER
				SEND IFC AND TAKE CHARGE
				SEND REN
				RESPOND TO SRQ
				SEND INTERFACE MESSAGES

The three handshake wires are:

NRFD	(NOT ready for data)
NDAC	(NOT data accepted)
DAV	(data valid)

ATN indicates that the controller is in charge and, when true (i.e. Lo),
ALL devices connected to the bus MUST listen.

REN allows all devices that have been addressed to go to remote.

IFC tells all device interfaces to clear down within 100µs.

SRQ is generated by TALKERS/LISTENERS requiring some form of attention from the controller.

EOI is used to indicate that the byte on the data bus is the last byte of the message. The EOI line is also used by some controllers (not the ATE) to perform a parallel poll.

There are five types of statement available for IEEE control from the ATE:

IEEE OUT
IEEE IN
IEEE CMD
IEEE SRQ
IEEE XFER

Examples of the syntax, and the specifications for each of these are given in the next section.

9.8.1 Syntax and specifications

IEEE OUT L4(/3;5), (N3,)("HELLO,")('OA OD')

a b c d

- a is the listener primary address (in this example, 4). Only one is allowable and it must lie between 1 and 62.
- b is a secondary address (defaultable) for the same listener, allowing expanded capability.
- c extra listener address. More than one listener may be addressed at the same time, up to a maximum of 14. Each extra listener can have an associated secondary address.
- d the output string can be :
 - i a string store (e.g. N3)
 - ii an ASCII immediate string ("HELLO")
 - iii a hexadecimal immediate string ('OA OD')
 - iv a concatenation of two or more of the above.

Strings i, ii and iii can each have an associated multiplier.

The output string is sent.

EOI is asserted true with the last byte.

UNL is sent to the bus.

If the Listener(s) require an EOS character, this must be specifically imbedded into the output string.

IEEE IN T4(/3),(MAXT 100ms),(EOS'ØA',)N4

a b c d

a is the talker address. Only one is allowed (in this example, 4) but expanded capability can be provided by a secondary address(/3) which is defaultable.

b is the time out, which is defaultable [see d) below and section 9.0.5].

c is optional, defining the end of string character (hexadecimal or ASCII). Single character only.

d is the destination store(s): up to 2 can be specified (i.e. a maximum of 240 bytes).

The input string is received.

The input is terminated if:

- a) EOI is received, or
- b) EOS character (optional) is received, or
- c) RESET is pressed, or
- d) time out (optional) is reached. FØ is set if time out occurs.

UNT is then sent to the bus.

IEEE CMD

Seven commands are available:

- 1) IEEE CMD, DCL Device Clear (all)

At the completion of a test program any externally controlled stimuli, e.g. Power Supplies, must obviously be turned off. However, if a test run is aborted half way, by pressing Reset or an early END statement, specific commands to these stimuli cannot be sent. However, in the same manner that Test Reset is sent down the Test Bus to close all ATE modules down, a Device Clear is sent down the IEEE 488 bus and it is important that all devices respond to it. Many do not. Therefore, when

using IEEE 488 stimulus instruments, their Interface Function Implementation list should be inspected for the inclusion of DC1 or DC2, (this list includes features like L1, AH1). It must be determined that the instrument responds in the desired way to Device Clear. Note that Feature DC0 means that it does not respond to Device Clear.

By way of example, Hendry 350 and 350.1 Bus Couplers claim: AH1, L1, DC1, E1; and state that "Device Clear... sets outputs to zero". This is satisfactory.

On the other hand, Kepco SN488 features: AH1, L1. Therefore this cannot be safely recommended.

- 2) IEEE CMD, GET, L4(/3;2) Group Execute Trigger (those addressed)
- 3) IEEE CMD, IFC Interface Clear (IEEE bus) (all)
- 4) IEEE CMD, LLO Local Lockout (all)
- 5) IEEE CMD, SDC,L4(/3;2) Selected Device Clear (those addressed)
- 6) IEEE CMD, GTL,L4(/3;2) Go To Local (those addressed)
- 7) IEEE CMD, LOC Go To Local (all). This disasserts REN for 100-150 μ s.

Note. GTL does not clear a device from Local Lockout state, but LOC does.

- a) If the command is an addressed command, the devices are addressed to listen.
- b) The command is sent.
- c) If listeners were present, UNL is sent.

IEEE SRQ (T4(/3;5),N4) Service request

The bus is tested for SRQ active and, if true, F1 is set.

If SRQ is true and devices have been specified, a serial poll is conducted to the bus. (If no devices have been specified, a serial poll is not conducted).

The primary address (4 in the example above), secondary address (3), and device status byte of all requesting devices are put into the specified string store. If there is no secondary address, FF is inserted.

A typical N4 could be:

$\emptyset 6$ / $\emptyset 4$ $\emptyset 3$ 42 / $\emptyset 5$ FF 21 /
 (1st device) (2nd device)

where

$\emptyset 6$ is the length byte
 $\emptyset 4$ and $\emptyset 5$ are primary addresses
 $\emptyset 3$ is a secondary address
 FF denotes no secondary address
 42 and 21 are status bytes

IEEE XFER T4(/3), L5(/2;1) (,MAXT 100ms)(,EOS ' $\emptyset A$ '))

The listeners are addressed and the bus checked for active listeners.

The talker is addressed.

The transfer is allowed to go.

Control is regained when:

- a) EOI is seen, or
- b) EOS character (optional) is seen, or
- c) time out (optional) is reached. F $\emptyset$ set if time out occurs before the transfer starts, F1 if it occurs before completion, or
- d) RESET is pressed.

UNL and UNT are sent to the bus.

NOTE:

If at any time RESET is pressed, REN will be disasserted for 100-150 μ s.

IFC will be sent true for 100-150 μ s. Device Clear (DCL) will be sent.

For further information on String Variable Stores, refer to section 9.9.20.

9.9 TEST STATEMENT SYNTAX & SPECIFICATIONS

The test facilities available are each described in two parts. The first, Statements, shows the correct format and sequence for the input of key-words and associated parameters. The language used is often self-explanatory but, in the second part - Specifications - the terms are described in some detail.

Capital letters (e.g.) MEAS are actual keyboard operations

Small letters (e.g.) range appear in places where the appropriate alphanumeric characters will be substituted

Optional statements or parameters are shown in brackets, such as:

[EB,]* or (S).* In these instances, unless an input is made for these fields, the system will use default values, or calculate them at run time.

Commas are shown as field separators. Except for the first one in a line, these can usually be omitted or replaced by spaces. However, if this is done, the reporting system for syntactical errors may indicate an incorrect likely fault.

Spaces can safely be inserted between the parameter key-words and their values.

If a display or list of statements is requested via the Functional Line Editor, these will be presented in a standardised format, irrespective of the input form.

Note that a character may have different meanings according to the context, e.g. C may signify Capacitor, Collector.

For clarity, the statement field (e.g. S10, S20, S30 etc) has been omitted from the beginning of each line.

All 'val' (value) entries may be either real numbers or memory references, e.g. M4. Real numbers must be within the allowed range for the particular statement and may be in any allowed multiplier form i.e. $1M = 1000U = 1E-3 = 0.001$.

* [] Statements within square brackets overwrite default conditions.

* () Units within parentheses may be defaulted. Note that for PIN(S) 'S' must be supplied for differential pins and omitted for single pins.

9.9.1 VERIFY Statements (Check value against limits)

Current (DC)

VER DCI,SHval(OHMS),[ITval(S),]PIN(S)pinrefs,HLval(A),LLval(A)

Voltage (DC)

VER DCV,[ITval(S),]PIN(S)pinrefs,HLval(V),LLval(V)

Voltage (AC)

VER ACV,[ITval(S),]PIN(S)pinrefs,HLval(V),LLval(V)[,EB]

Frequency

VER FREQ,POLpol,[CYCival,][TLval(V),]PIN(S)pinrefs,HLval(HZ),LLval(HZ)[,VRrange]

Event count

VER EV,POLpol,[GTval(S),][TLval(V),]PIN(S)pinrefs,HLval,LLval[,VRrange]

Peak voltage

VER PK,[GTval(S),]PIN(S)pinrefs,HLval(V),LLval(V)[,EB]

Trough voltage

VER TR,[GTval(S),]PIN(S)pinrefs,HLval(V),LLval(V)[,EB]

Period

VER PER,POLpol,[TLval(V),]PIN(S)pinrefs,HLval(S),LLval(S)[,VRrange]

Pulse width

VER PW,POLpol,[TLval(V),]PIN(S)pinrefs,HLval(S),LLval(S)[,VRrange]

For in-circuit statements, refer to section 9.9.16.

-
- * All numerical values of quantities (as distinct from addresses) can be specified in two ways:
- a) input via keyboard, b) by reference to a memory (containing a previously specified number).
- Numbers (assumed positive unless prefixed by -) can be expressed using 4 characters and decimal point to establish magnitude, with a multiplier or exponent term appended if necessary. For details of the alternative formats accepted by the system, refer to 'Numerical Storage Capacity' (sub-section of 9.0.1). Dimensions (units, e.g. volts, Hz, amps, seconds) are assumed automatically.

9.9.2 VERify Specifications

The truth table of automatically applied flag settings is on page 9-79.

DCI	Current, determined from V developed across SHunt R ($I=V/R$).	
DCV	d.c. voltage	} Range determined from numerically highest HL or LL, e.g. HL 1V, LL -10V.
ACV	a.c. voltage	
FREQ	frequency	
EV	event count, i.e. number of threshold crossings in defined direction. Maximum 65,535. ($2^{16}-1$).	
PK	peak	} Absolute voltage values. Only usable for repetitive signals.
TR	trough	
AC	average-responding, rms calibrated assuming sinewave shape.	
PER	period, i.e. time between consecutive similar transitions (one cycle). Maximum 8 seconds.	
PW	pulse width: time between consecutive dissimilar transitions. Maximum 8 seconds.	
SH	shunt value of resistor across which system measures voltage for derivation of DCI. (Resistor in series with UUT, preferably at low voltage (earthy) end of current path). SHval assumed ohms*.	
IT	integration time assumed seconds*. Programmable 100 μ s to 20ms. Default: 20ms. (16.67ms for 60Hz).	
pinref(s)	PIN is single-ended. PINS are differential, A first, C second.	
HL**	high limit (to precede LL). Units are basic* - ohms, seconds, amps, volts, Hz.	
LL**	low limit. Units as for HL.	
VR	voltage range. Number (1-8) representing one of binary series of voltage ranges from 0.5V to 64V (list in section 9.9.23).	
EB	extended bandwidth (up to 200kHz).	
CYCiva1	number (<u>integer value</u>) of cycles - 16 or 256 - from which frequency is computed.	
TL	threshold level, value assumed volts*. Applicable to all analog timing operations (frequency, event, pulse, period). By default, the timing circuit is a.c. coupled (see 'Timing Measurements' on page 9-8).	
POL	polarity: POS or NEG. By default NEG is assumed.	
GT	gate time, value assumed seconds*.	

* See footnote on previous page

** One of these may be defaulted

9.9.3 MEASURE Statements (Read value into a system store)

Current (DC)

MEAS DCI,SHval(OHMS),[ITval(S),]PIN(S)pinrefs[,VRrange][,INTOstore]

Voltage (DC)

MEAS DCV,[ITval(S),]PIN(S)pinrefs[,VRrange][,INTOstore]

Voltage (AC)

MEAS ACV,[ITval(S),]PIN(S)pinrefs[,VRrange][,EB][,INTOstore]

Frequency

MEAS FREQ,POLpol,[CYCival,][TLval(V),]PIN(S)pinrefs[,VRrange][,INTOstore]

Event count

MEAS EV,POLpol,[GTval(S),][TLval(V),]PIN(S)pinrefs[,VRrange][,INTOstore]

Peak voltage

MEAS PK,[GTval(S),]PIN(S)pinrefs[,VRrange][,EB][,INTOstore]

Trough voltage

MEAS TR,[GTval(S),]PIN(S)pinrefs[,VRrange][,EB][,INTOstore]

Period

MEAS PER,POLpol,[TLval(V),]PIN(S)pinrefs[,VRrange][,INTOstore]

Pulse width

MEAS PW,POLpol,[TLval(V),]PIN(S)pinrefs[,VRrange][,INTOstore]

- * All numerical values of quantities (as distinct from addresses) can be specified in two ways:
- a) input via keyboard, b) by reference to a memory (containing a previously specified number).
- Numbers (assumed positive unless prefixed by -) can be expressed using 4 characters and decimal point to establish magnitude, with a multiplier or exponent term appended if necessary. For details of the alternative formats accepted by the system, refer to 'Numerical Storage Capacity' (sub-section of 9.0.1). Dimensions (units, e.g. volts, Hz, amps, seconds) are assumed automatically.

9.9.4 MEASure Specifications

DCI	Current, determined from V developed across SHunt R ($I = V/R$).
DCV	d.c. voltage.
ACV	a.c. voltage.
FREQ	frequency.
EV	event count, i.e. number of threshold crossings in defined direction. Maximum 65,535. ($2^{16}-1$).
PK	} Absolute voltage values.
TR	
	trough } Only usable for repetitive signals.
AC	average-responding, rms calibrated, assuming unity form-factor.
PER	period, i.e. time between consecutive similar transitions (one cycle). Maximum 8 seconds.
PW	pulse width: time between consecutive dissimilar transitions. Maximum 8 seconds.
SH	shunt value of resistor across which system measures V for derivation of DCI. (Resistor in series with UUT, preferably at low voltage (earthy) end of current path). SHval assumed ohms*.
IT	integration time, assumed seconds*. Programmable 100µs to 8 seconds. Default: 20ms.**
pinref(s)	PIN is single-ended. PINS are differential, A first, C second.
VR	voltage range. Number (1-8) representing one of binary series of voltage ranges from 0.5V to 64V (list in section 9.9.23).
INTO	into store Mx where x = 0-19.***
EB	extended bandwidth (up to 200kHz).
CYCival	number (<u>integer</u> <u>value</u>) of cycles - 16 or 256 - from which frequency is computed.
TL	threshold level, value assumed volts*. Applicable to all analog timing operations (frequency, event, pulse, period). By default, the timing circuit is a.c. coupled (see 'Timing Measurements' on page 9-8).
POL	polarity: POS or NEG. By default, NEG is assumed.
GT	gate time, value assumed seconds*.
store	full store reference e.g. M4 required. Default is M0.

* See footnote on previous page

** 16.67ms for 60Hz.

*** By default, stored in M0.

9.9.5 ADJUST Statements (Adjust component and display pictorial value and limits)

Voltage (DC)

ADJ DCV,[ITval(S),]PIN(S)pinrefs,NOMval(V),HLval(V),LLval(V),CREFref

Voltage (AC)

ADJ ACV,[ITval(S),]PIN(S)pinrefs,NOMval(V),HLval(V),LLval(V),[EB,]CREFref

Frequency

ADJ FREQ,POLpol,[CYCival,][TLval(V),]PIN(S)pinrefs,NOMval(HZ),HLval(HZ),LLval(HZ)[,VRrange],CREFref

Event count

ADJ EV,POLpol,[GTval(S),][TLval(V),]PIN(S)pinrefs,NOMval,HLval,LLval[,VRrange],CREFref

Peak voltage

ADJ PK,[GTval(S),]PIN(S)pinrefs,NOMval(V),HLval(V),LLval(V),[EB,]CREFref

Trough voltage

ADJ TR,[GTval(S),]PIN(S)pinrefs,NOMval(V),HLval(V),LLval(V),[EB,]CREFref

Period

ADJ PER,POLpol,[TLval(V),]PIN(S)pinrefs,NOMval(S),HLval(S),LLval(S),CREFref

Pulse width

ADJ PW,POLpol,[TLval(V),]PIN(S)pinrefs,NOMval(S),HLval(S),LLval(S),[VRrange,]CREFref

- * All numerical values of quantities (as distinct from addresses) can be specified in two ways:
- a) input via keyboard, b) by reference to a memory (containing a previously specified number).
- Numbers (assumed positive unless prefixed by -) can be expressed using 4 characters and decimal point to establish magnitude, with a multiplier or exponent term appended if necessary. For details of the alternative formats accepted by the system, refer to 'Numerical Storage Capacity' (sub-section 9.0.1). Dimensions (units, e.g. volts, Hz, amps, seconds) are assumed automatically.

9.9.6 ADJust Specifications

DCV	d.c. voltage.
ACV	a.c. voltage.
FREQ	frequency.
EV	event count, i.e. number of threshold crossings in defined direction. Maximum 65,535. ($2^{16}-1$).
PK	peak
TR	trough
	} Absolute voltage values
	} Only usable for repetitive signals.
AC	average-responding, rms calibrated, assuming unity form-factor.
PER	period: time between consecutive similar transitions (one cycle). Maximum 8 seconds.
PW	pulse width: time between consecutive dissimilar transitions. Maximum 8 seconds.
IT	integration time, assumed seconds*. Programmable 100 μ s to 20ms. Default: 20ms.**
pinref(s)	PIN is single-ended. PINS are differential, A first, C second.
NOM	nominal value (not necessarily mid-way between HL and LL) for ease of reference. Accuracy not stated.
HL***	high limit (to precede LL). Units are basic* - ohms, seconds, amps, volts, Hz.
LL***	low limit. Units as for HL.
EB	extended bandwidth (up to 200kHz).
CYCival	number (<u>integer value</u>) of cycles - 16 or 256 - from which frequency is computed.
TL	threshold level, value assumed volts*. Applicable to all analog timing operations (frequency, event, pulse, period). By default, the timing circuit is a.c. coupled (see 'Timing Measurements' on page 9-8).
VR	voltage range. Number (1-8) representing one of binary series of voltage ranges from 0.5V to 64V (list in section 9.9.23).
POL	polarity: POS or NEG. By default NEG.
GT	gate time, value assumed seconds*.
CREFref	component reference. 1 to 6 characters inside " ".

* See footnote on previous page.

** 16.67ms for 60Hz.

*** One of these may be defaulted.

9.9.7 RELAY Statements (Open or close a relay on an RRM)

OP,pinlist

CL,pinlist

NOTE: A pinlist is a list of a least one pin reference of the form

24/A7

separated by commas.

Several Pin numbers may be associated with one module number by use of colons

e.g. OP, 24/A7:C4:A3,15/A4:A6

All relays of a module may be actioned by using an asterisk

e.g. CL,24/*

Note: OP and CL instructions must not be mixed on one line.

Relay connections are numbered in groups of three, e.g. A1:A2:A3, or C10,C11,C12 etc, where the central number (A2,C11) is the unswitched common line. For this reason statements must not specify opening or closing of connections to these points (see Test Interface diagram).

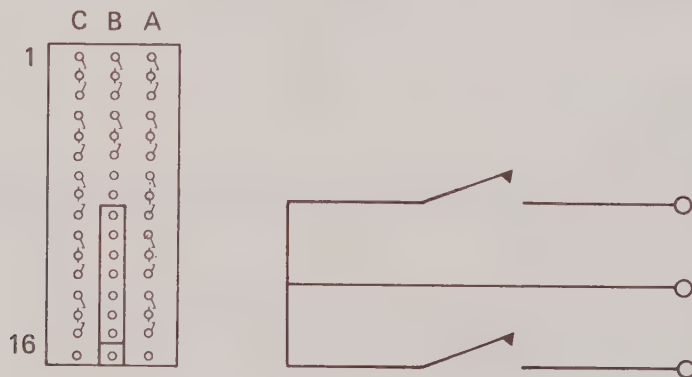


Fig. 9.8 Reed Relay Connections (TIC)

9.9.8 RELAY Specifications

Relay contacts (on TIC) are grouped in threes (1,2,3), (4,5,6) etc where the central pin in each group (2), (5) etc is common to the switched contacts (1 and 3), (4 and 6) etc. Relays are on A1-A15, C1-C15 and B1-B6. B9-B15 are test bus address pins. B16 is digital ground.

9.9.9 Truth Table of Automatically Applied Flag Settings
for VER/ADJ Statements

TEST RESULT	Transitory Flags on exit				Global Flags		
	GO	NOGO	NOGOHI	NOGOLO	FAIL	PASS	F99
GOOD	1	0	0	0	NO CHANGES		
HIGH	0	1	1	0	1	0	1
LOW	0	1	0	1	1	0	1

9.9.10 STIMULUS/POWER SUPPLY Statements (Set up, switch on, switch off
a stimulus module)

Power Supplies (PSM,PCM)

SU PSU,PSpsuref,TOval(V),[ILIMval(A)]

APP PSU,PSpsuref

REM PSU,PSpsuref

A PSU is controlled and interfaced by the PSM (or PCM) and each PSM (PCM) controls two identical supplies. Each supply is fully floating and programmable 0-32V in 10mV steps (2mV PCM) and 0-1 Amps in 5mA steps (not PCM). Alternatively, current limiting may be controlled with resistors on the test connector (see section A2 for characteristic). Default current is 1.1 Amp with PSM. 1.25A with PCM.

psuref is (module number)/A or C. Module number may be 1, 2 or 3.

One or two PSM (PCM)s may be fitted as standard and a third fitted as non-standard.

DC Stimuli (VIM)

SU DC,MODdcref[,VOLTSval(V)][[,IPOSval(A)][[,INEGval(A)]]

APP DC,MODdcref

REM DC,MODdcref

Sine Wave Stimulus (SSM)

SU SINE,MODinteger,FREQval(HZ),RMS val(V)

APP SINE,MODinteger

TRIM SINE,MODinteger,PIN(S)pinrefs,TO val(V)

REM SINE,MODinteger

Pulse Stimulus (PGM)

SU PULSE,MODinteger,MKVval,SPval[,PULSESval][,MKVval][,SPVval]

SU GTO,MODinteger[,MKVval][,SPVval]

SU CTA,MODinteger[,MKVval][,SPVval]

SU CTB,MODinteger[,MKVval][,SPVval]

APP PGM,MODinteger

START PULSE,MODinteger[,FROMcty],POLpol

STOP PGM,MODinteger

REM PGM,MODinteger

General

REM ALL (not A8000)

9.9.11 STIMULUS/POWER SUPPLY Specifications

PSM power supply module(s). One or two can be fitted, each capable of controlling two PSUs.

PSU Two floating outputs, 0 to 32V. Max. current 1.1A (or as determined by program or resistor)* at all voltages. Outputs can be series-connected for higher voltages or parallel-connected to increase current capability.**

dceref (Module number)/A or C

- * On Hendry power supplies, for current limit up to 1A, use set resistor of value given by

$$R_I = \frac{249I_{LIM} + 28.67}{1.2417 - I_{LIM}}$$

where R is in ohms and I is in amps. A graph of R_I vs I_{LIM} is in Appendix A.2.

- ** If two or more PSU outputs are connected in parallel to increase the current output to 2A or more, it is essential to fit programming resistors of $\frac{1}{2}$ 1200 ohms (equivalent to 1A current), or specify $I_{LIM}=1$, to all linked supplies. Failure to do so will cause one or more supplies to limit at 25% above rating and may blow the fuse on the Hendry power supply.

psuref 1/A, 1/C, 2/A or 2/C. [3/A, 3/C for third pair].

DC,MOD voltage/current module (VIM) providing dual programmable force stimuli. VOLTS up to $\pm 12.8V$. IPOS & INEG up to 128mA, default 0A (do not default both).

SINE from sine source module (SSM). FREQ range 20Hz to 200kHz. RMS output 0 to 8V (2mV increments). Quiescent (unused) output impedance : $1G\Omega$.

PULSE from pulse generator module (PGM). MK & SP times: 1 μ s to 4.1943s (dependent on resolution). PULSES: 1 to 65,535 (default=continuous). MKV & SPV: -12.8V to +12.8V.

GTO gate time output

CTA control line A } for triggering PGM output to coincide with

CTB control line B } the integration time of a measurement.

cty one of CTA, CTB and GTO.

PGM SOFTWARE SYNTAX - EXPLANATORY NOTES

STATEMENT STRUCTURE

- 1 SU PULSE, MOD X, MK YY, SP YY, [PULSES N,] [MKV ZZV,] [SPV ZZV] sets up the PGM as a pulse generator.
X defines the module address.
YY Mark/Space time range is 1 μ s - 4.195s.
PULSES Parameter defines the number of pulses; if defaulted, continuous pulse train.
N No. of Pulses 1 - 65,535.
MKV, SPV Set the mark/space level of the main output. If defaulted the previously set level is left. If no previously set level, zero is set.
ZZ Output voltage -12.8 to +12.8.
- 2 SU CTA, MOD X, [MKV ZZV,] [SPV ZZV]
 CTB
 GTO
 sets up the PGM to mimic CTA, CTB or GTO all parameters as above.
- 3 APP PGM, MOD X
 Turns on O/P.
- 4 REM PGM, MOD X } REM instant if stopped or on continuous. Waits to end of pulse
 Turns off O/P. } train if train length programmed. Always initialises PGM.
- 5 STOP PGM MOD X
 Stops PGM; reverts to space. Asynchronous.
- 6 START PULSE, MOD X, [FROM CTA, POL POS.]
 CTB NEG
 GTO
 This starts the previously set up pulse generator.

FROM PARAMETER If this is defaulted, the pulse generator is started immediately. Otherwise the PGM is started on next, defined edge of CTA, CTB or GTO.

USING PGM STATEMENTS

If a new SU command is issued after a pulse train is running, the PGM will be reconfigured immediately and the O/P stopped if the pulse train was continuous. If a number of pulses had been specified the system will wait until these are finished before reconfiguring.

If a new START command is issued after a pulse train is running, the message INVALID START IGNORE will be output and the system will continue if the pulse train was continuous. If a number of pulses had been specified the system will wait and then implement the start statement.

The SU CTA etc. statements require no START as they only mimic the appropriate line.

The STOP statement stops the pulse train if selected and configures the O/P to the SPACE level. The STOP command is totally asynchronous. If a STOP command is issued but the PGM is already stopped, the message INVALID STOP IGNORE is output and the system continues.

The APP statement applies the output; this is totally asynchronous.

The REM statement tri-states the output. If a pulse train of defined length is running, the system waits until it finishes before tri-stating.

The range selected by the system depends on the maximum pulse width specified. Eight ranges provide resolutions from 0.5 μ s to 64 μ s, and maximum pulse width on a given range is resolution x 65,536. Minimum is resolution x 2. See Appendix A8 for PGM Specification.

NOTE The output rest state of the PGM, if applied, is SPACE. When triggered the MARK state is immediately entered.

9.9.12 CONTROL Statements

Delay

DLY,val(S)

Compare value against limits

COMP,f.p.store,HLval,LLval

COMPFAIL,f.p.store,HLval,LLval

SCOMP (See section 9.9.20).

Set/Unset Flags

FLAG,flag,[IFflag]

UNFLAG,flag,[IFflag]

Maths Operations

LET,f.p.store=expression

String Store Operations

SLET }
SLEN } (see section 9.9.20).

Program Branching

GOIF,flag,Sival

GOTO,Sival

Execute Statement(s)

EX,Sival[,*ival]

I/O Operations (SIO)

IN,Pinteger,stores

OUT,Pinteger,stores and/or strings

Display Message

DISP,stores and/or strings

Output to ticket printer

PRINT,Winteger,stores and/or strings

IEEE 488 Bus Interactions

IEEE OUT]
IEEE IN]
IEEE CMD]
IEEE SRQ]
IEEE XFER] (see section 9.8.1)

Modified Running Conditions

MODE,FAST NORMAL UNCAL (not A8000)
END

9.9.13 CONTROL Specifications

DLY	delay 100µs - 8s.
COMP	compare stored floating point number with high limit and/or low limit values (either or both limit fields allowed). If stored value lies outside limit(s), then NOGO & NOGOHI/NOGOLO (as appropriate) are set.
COMPFAIL	as COMP, except that the result of a value lying outside limits is FAIL being invoked and F99 being set.
SCOMP	compare whole or part contents of a string variable store with whole or part contents of another string variable store, or floating point store, or string, etc, flags being set as for COMP.
FLAG	any flag may be set independently or as a result of any other flag being set.
UNFLAG	refers only to F0 to F7 and F99, any of which can be independently reset as a result of any flag being set.
LET	floating point store, M0 to M19, to become equal to some expression. "expression" [from "Let, store=expression" above] is a normal algebraic expression with optional brackets ().

In the absence of brackets, operations are performed in the following sequence:

functions	e.g. SQRT or, with extended maths facility*, sin,cos,tan,log. Function arguments should be enclosed in brackets.
signs	+ or - ascribed to a number (e.g. -27.3) or to an exponent (e.g. 12.7E-6). If the signs do not immediately follow the =, the operator and argument must be enclosed in brackets.
powers**	e.g. $(6.32)^2$ entered as (6.32^2)
multiplication (*) and division (/)	
addition (+) and subtraction (-)	

When brackets are present, operations within them are performed first.

For details of the High Speed Extended Maths facilities, see Appendix A14.

SLET	assign a string to a string variable store, e.g. N0 to N4.
SLEN	assign to a floating point store the character count from a string variable store.

* Optional extra on A8000; standard on A8315.

** Only with extended maths.

GOIF	automatically routes to a direct address or an indirect address (related to a floating point store) as a direct result of any flag being set. (S = executable Statement).
GOTO	automatically routes to a direct address or an indirect address (related to a floating point store). (S = executable Statement).
EX	executes the statement number directly or indirectly defined by the number of times defined in the [*ival] field. (S = executable Statement).
IN	an input via SIO port 1 or 2 into either a floating point, a string, or a string variable store.
OUT	an output via SIO port 1 or 2, contents of stores, and/or strings. (P1 = VDU).
DISP	an output to the operator display, contents of stores or strings, acts as a break point. Program will not be continued until PROCEED is pressed.
PRINT	size of character on ticket printer determined by W5, W10, W20 or W40 (number of characters per line). (W = width).
IEEE	the five types of statement available for control are OUT, IN, CMD, SRQ and XFER (see section 9.8.1).
END	logical end to program sequence (can appear anywhere in a program).

FAST MODE

This running mode saves approximately 10-15ms per statement by not displaying the statement number or measured value. This mode is invoked by

S50 MODE,FAST

which can occur anywhere within the test program. On encountering this statement, the message

RUNNING FAST MODE

will be displayed, and no further access is made to the display, except for explicit display statements, adjusts and program failures. The machine can be made to run normally by invoking either PRINT ALL or SINGLE STEP. Note that fast mode is not reinstated after cancelling either of these modes (unless another MODE FAST encountered). Normal mode may also be invoked under program control by:

S100 MODE,NORMAL

When using FAST MODE in conjunction with an ADD program, a MODE FAST statement must occur after the DIAGBEGIN to avoid slowing down of signature capture during fault tracing. If a MODE FAST statement occurs at the beginning of the program, simply repeating this statement after DIAGBEGIN will allow ADD capture to run at its normal speed. This is a consequence of the ADD/MODE facilities' effect on system flags and does not affect Passing program runs if not adhered to.

If this condition is not followed, there will be no effect on program runs not invoking ADD.

UNCAL MODE

When performing digital tests or simple functional tests which do not require the full spec. of the system, a MODE can be invoked which does not recalibrate the ACM at the start of every test program run. This is done by again using the MODE statement e.g.

```
S1      MODE,UNCAL
```

This mode will save approximately 2 seconds on a test program run.

NOTE: The MODE UNCAL statement must be the first statement in the program, i.e. it must have the lowest statement number; this rule also includes declaratives. It is good discipline, to ensure this, to make the MODE,UNCAL statement as S1.

More than one mode may be concatenated into the same statement; these are acted upon in a sequential manner from left to right, e.g.

```
S1      MODE,UNCAL,FAST
```

Example of CONTROL Statements for sub-routines

The sample program below shows the use of memory-referenced jumps and executes to establish a loop for a sequence of similar operations.

S5301 LET M11=1	% C PIN REF NO.
S5302 LET M12=1	% LTM BANK SELEC
S5303 LET M13=7001	% MEAS STATEMENT
S5400 SET,#TDATA=F73,#C=M12,#A=0.CLK 7/A14	% SET UP FOR PIN
S5410 LET M14=5450	% RETURN ADDRESS
S5420 GOTO S6000	% MEAS ROUTINE
S5450 SET,#A=7FFF,#C=7FFF	% XTALK TEST
S5460 LET M14=5500	% RETURN ADDRESS
S5470 GOTO S6000	% MEAS ROUTINE
S5500 COMP M11,LL 7	% LAST PIN?
S5550 GOIF GO S8000	% YES,SO OUT
S5560 LET M11=M11+1	% NEXT PIN
S5570 LET M12=M12*2	% NEXT LTM PIN
S5580 LET M13=M13+1	% NEXT MEAS STA
S5620 GOTO S5400	% LOOP BACK
S6000 MEAS DCV,PIN33/C15,VR1,INTO M3	% VOUT
S6010 EX SM13	% MEAS INPUT V
S6020 LET M5=(M3-M2)/M4	% PIN GAIN
S6030 COMPFAIL M5,HL-99.5M,LL-100.5M	% GAIN TEST
S6060 GOTO SM14	% RETURN

```

S7001 MEAS DCV PIN4/C1,VR4,INTO M4          % TABLE OF
S7002 MEAS DCV PIN4/C2,VR4,INTO M4          % MEASURE
S7003 MEAS DCV PIN4/C3,VR4,INTO M4          % STATEMENTS
S7004 MEAS DCV PIN4/C4,VR4,INTO M4          % FOR C PINS.
S7005 MEAS DCV PIN4/C5,VR4,INTO M4
S7006 MEAS DCV PIN4/C6,VR4,INTO M4          % INPUT VOLTS
S7007 MEAS DCV PIN4/C7,VR4,INTO M4          % INPUT VOLTS

S8000.....

```

Information to the right of the % signs appears only during program entry and, if a record is required, the ticket printer must be operated before the statements are scrolled off the VDU. The information is a COMMENT mode and is not part of the program.

The GOTO operations annotated 1, 3 and 5 are branches to a direct address.

The GOTO operations annotated 2 and 4 are branches to a memory-referenced address.

S6010 initiates a one-line subroutine, after which the program continues with S6020.

S7001 to S7007 constitute a table of memory-referenced statements. (They are called up via M13).

9.9.14 TIMER Statements

Timer statements are compound statements requiring at least three separate statements to perform the task. The ATE system contains two internal timing lines, CTA and CTB, on which a variety of measurements may be made. There are three main components which must be in correct order:

- 1) CONnect signal source ONTO counter timer line. Source may be either an LTM pin or a TIM pin (for which Threshold Level and Noise Immunity may be specified).
- 2) PREPare TIMER, MODE whatever one wants to measure, e.g. FREQ CTA, ctA to ctB (AB), plus POLarity fields (default = negative). This statement arms the timer.
- 3) MEAS / VER TIMER against limits,collects the answer and processes it into either a memory store or verifies against limits.
- 4) DISconnect source from timer line is optional and dependent on the rest of the program.

Connect

```

CON LTM,PINpinref,ONTOctx
CON TIM,PINpinref(,TLVAL) (,Nival),ONTOctx

```

Prepare

```

PREP TIMER,MODEmm,STARTpol,STOPpol
PREP TIMER, MODE FREQ CTA (,POLPOS)      }
  "      "      "      " CTB (,POLNEG)    }      FREQ MODE

```

continued.....

Prepare (continued)

PREP TIMER, MODE EV	CTA (,POLPOS)	}	EVENTS MODE
" " " "	CTB (,POLNEG)		

Measure/Verify

MEAS TIMER(,INTOstore)
 VER TIMER,HLval(S),LLval(S)

Disconnect

DIS LTM,MODinteger
 DIS TIM,MODinteger
 DIS CTA
 DIS CTB

9.9.15 TIMER Specifications

a Any CONnect statement will automatically disconnect any LTM or TIM previously connected to that CT line before connecting the new pin. TL: threshold level. NI: noise immunity.

b Certain measurements require the use of the TIMER and CTA by the system.

These are:

SU SINE
 SU PSU (PCM ONLY)
 ANY ANALOG VOLTAGE OR PERIODIC MEASUREMENT
 ANY IN-CIRCUIT MEASUREMENT

If any one of these is executed but the CTA line is occupied, a programming error is assumed.
 The message :

CTA IN USE, MOD XXX (where XXX is the TIMER or LTM connected) will be printed and the program terminated.

c Similarly if a PREP TIMER has not been followed by a MEAS (or VER) TIMER and one of the above statements is encountered, the message:

TIMER IN USE will be printed and the program terminated.

d If a MEAS (VER) TIMER statement is executed with no previous PREP TIMER, the message:

TIMER NOT PREPARED will be printed and the program terminated.

FREQ MODE MAX FREQ 1.5MHz

The PREP TIMER simply arms the system. The input freq is not measured until a MEAS (VER) TIMER statement is encountered. Thus the input waveform must be continuous at the MEAS (VER) TIMER statement, when the system performs an autoranging frequency measurement.

NOTE As the measurement is only performed at the MEAS (VER) TIMER statement, repetitive frequency measurements can be done by simply only re-executing the MEAS (VER) TIMER statement. This only works provided no other PREP TIMER statement or any of the TIMER dependent measurements are executed since the last PREP TIMER, MODE FREQ.

NO PULSES TIME OUT and F0 as per analog Frequency Measurements (see section 9.0.5).

EVENTS MODE MAX EVENTS 2^{16} ($2^{16}-1$) [4,294,901,760].

The PREP TIMER statement arms the counter and then opens the GATE (GTO). The MEAS (VER) TIMER statement closes the GATE and reads the counter. A typical program could be:

S10 CON LTM PIN 8/A1 ONTO CTB	←	SELECT I/P LINE
S20 PREP TIMER, MODE EV CTB,POLPOS	←	PREP TIMER, GATE OPEN
S30 SET CLK 8/A14	←	TRIGGER EVENTS FROM UUT
S40 DLY 100ms	←	WAIT FOR EVENTS TO FINISH
S50 VER TIMER HL6400, LL6399	←	CLOSE GATE, FETCH RESULT

Repeated execution of the MEAS (VER) TIMER statement in this mode and in the Timer Interval Mode (AA,AB etc) will always yield the same answer. Any successive measurement requires a new PREP TIMER statement.

NOTE As the result of a TIMER measurement is dependent upon the PREP TIMER statement, the limits entered in the VER TIMER statement must be unit-less.

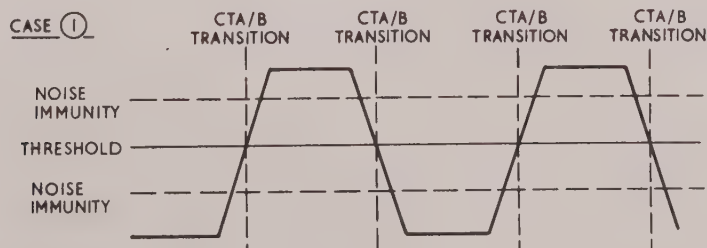
DIS CTA, DIS CTB

These statements automatically disconnect the LTM or TIM connected to the specified CT line.

Levels for Timing Measurements (TIM only)

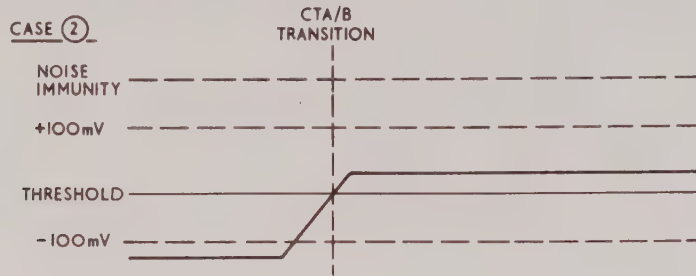
Timing measurements are dependent on the static level and the amplitude of the test signal as related to programmed threshold and noise immunity levels. This requirement is described and illustrated for three types of measurement. Default threshold = 0V, Default Noise Immunity = 100mV.

Case 1 A repetitive signal must cross the programmed threshold level, by the programmed noise immunity, in both directions.

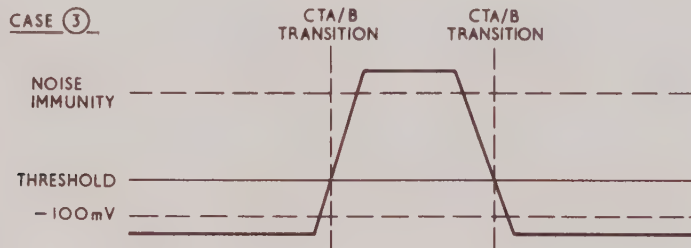


9-90

Case 2 With a one-off occurrence, the static level of the signal must be at least 100mV outside the threshold low.



Case 3 To measure the width of a single pulse, the static level must be at least 100mV outside the threshold low, and the signal must cross the threshold by the noise immunity before reverting to the static level.



Note In cases 2 and 3, the static level must be correct at the CON TIM statement and must not change before the PREP TIMER statement.

9.9.16 ZTM Statements

Tristate/Guard

TRI pinlist

GRD pinlist

Resistor

VER R,DRx/by,SNx/by,HLval,LLval,[DLYval,]CREFref

Capacitor

VER C,DRx/by,SNx/by,HLval,LLval,CREFref

Polarised capacitor

VER C ELEC,DRx/by,SNx/by,HLval,LLval,CREFref

Inductor

VER L,DRx/by,SNx/by,HLval,LLval,CREFref

Impedance

VER Z,FREQval,DRx/by,SNx/by,HLval,LLval,CREFref

Diode

VER D,DVm,ANx/by,KAx/by,[MAXVval,][DLYval,]CREFref

Biode

VER B,DRx/by,SNx/by,[MAXVval,][DLYval,]CREFref

Transistor (PNP)

VER PNP,[ONLY dir,]Ex/by,Bx/by,Cx/by,[DLYval,]CREFref

Transistor (NPN)

VER NPN,[ONLY dir,]Ex/by,Bx/by,Cx/by,[DLYval,]CREFref

NOTE: ALL LINEAR STATEMENTS (WITH LIMITS) RETURN THE MEASURED IMPEDANCE/RESISTANCE
OF THE COMPONENT IN STORE MØ.

9.9.17 ZTM Specifications

TRI pinlist	Any list of ZTM pins. When declared as a ZTM, the test module will have its pins		
GRD pinlist	in the tristate mode unless defined otherwise. If defined as an ITM, then the only possible combination of pins is Guard, Drive or Sense. When pins are called on a particular statement as Drive and Sense, after that test they will revert to the condition applying prior to that test, i.e. Tristate or Guard. Examples of ZTM Statements are:		
	1)	TRI 7/A10:A12:C3, 8/A1:A2	
		where five pins are selected to be Tristate	
	2)	GRD ALL	
		where all pins are to be in the Guard mode.	
VER	A check to verify that the value of a passive component does not lie beyond HI and/or LO limit(s) defined in the program. Actual value will automatically be stored in MØ. In the case of active components (biodes, diodes, transistors) the equipment will verify that the characteristic lies within the limits of a model defined by the test system.		
DRx/by	Drive (point of stimulus application) where		
	x	=	module address
	b	=	pin bank (A,B or C)
	y	=	pin number
HL	High limit of component value	}	Either or both may be specified. (HL first if both specified).
LL	Low " " " "		
	Limits also define stimulus level, frequency (in the case of L & C) and detector sensitivity.		
DLY	Delay of up to 8 seconds may be specified where required to permit circuits to stabilise from the time of application of a stimulus to the time of sensing for a check. Units are seconds unless M appended (for milli).		
CREf	Component circuit reference number for identification.		
FREQ	In the case of impedances, the value is frequency-dependent and, in this context, it is necessary to define the frequency at which the impedance is to be tested. It is desirable to select the lowest frequency available for a given type of measurement.		
DVm	Drive; m will be A for anode or K for cathode.		
ANx/by	AN = Anode (x/by is location of anode pin : x,b,y as for DR).		

KAx/by KA = Cathode pin location.

MAXVval Maximum stimulus voltage. Programmed value up to 5V. By default, maximum drive voltage is 1V.

ref 1 to 6 characters inside double quotes " ".

Transistors (Bipolar PNP and NPN types)

E,B,C Emitter, base, collector (x,b,y as for DR)

ONLY Transistors are normally verified by a number of forward and reverse tests, driving emitter while measuring collector and base currents, and then driving collector while measuring emitter and base currents. Performance is assessed algorithmically from these results. However, circuit configurations sometimes invalidate either the forward or reverse set of tests, compelling the programmer to specify ONLY FWD or ONLY REV as appropriate. The basic principle of establishing the non-linear component of current is the same as that used for checking diodes (see section 9.5.9).

An example of ZTM statements commences at step S520 in section 9.6.2 (see page 9-43).

9.9.18 DIGITAL Statements

Each individual digital or logic test module may have specific voltages programmed for either a single threshold or for two threshold levels.

Modify digital sensing threshold(s)

SU SENSE,MODinteger,T0val(V) (In IPG, use of <ret> after T0 makes dual threshold available)

SU SENSE,MODinteger,H1val(V),L0val(V) (For dual threshold)

Allocate a name to a group of pins

DEF NAME,#name(t[,b])=pinlist

Digital Test/Stimulus Functions

SET,pinchange pinlist

SET,#name=action

SET,store=#name

SET,sample

SET,CLKpinref[*jval]

SET,list of two or more of the above components linked by use of commas.

9.9.19 DIGITAL Specifications

pinlist	linked list of pinrefs in msb to lsb (most to least significant bit)	
	e.g. 45/A1:A2:A3,120/C4	
	Note: 91/* actions or allocates all pins for LTM only	
	(in DEF NAME,pinA16=msb, pinA1=lsb)	
	In SET, ALL is also permissible as a pinlist	
name	ASCII literal string (max 6 chars)	
t	no of bits, must agree with pinlist	
b	number base of future pattern references	
	B	binary
	O	octal
	D	decimal
		default hexadecimal
pinchange	HI	
	LO	
	INV	invert
	SN	sense
	DR	drive
	X	ignore (don't care)
action	pinchange	
	pattern	i.e. a number of the appropriate base store
	PI	pattern increment
	PD	pattern decrement
sample	VER	sets pass/fail and go/nogos
	MEAS	sets go/nogos
	APP	actions changes to hardware - automatic at end of SET

SET CLK 7/A1 *5 will in fact change the level of that drive pin from its original state to its inverted state, and back again, a total of five times at a rate equivalent to 50kHz. [*ival can be a store].

Examples

```
DEF NAME,#DATA10(16)=56/*
DEF NAME,#ABUS(8)=29/A8:A7:A6:A5:A4:A3:A2:A1
SET,HI ALL,DR 43/A6,#ABUS=3F,CLK 29/C12,VER
SET,LO 43/A6,#ABUS=M16,APP,#ABUS=PI
SET,M4=#ABUS,#DATA10=DR,MEAS
```

9.9.20 String Store Statements

Allocate list contents to a string store

SLET, store = I/O list

Compare two strings

SCOMP, string , string

Equate store to length of string

SLEN, store = I/O list

I/O Operations (SIO)

IN,Pinteger,stores

OUT,Pinteger,stores and/or strings

Display Message

DISP,stores and/or strings

Output to ticket printer

PRINT,Winteger,stores and/or strings

9.9.21 String Store Specifications

store = string store / f.p. store

string store = N0 N1 N2 N3 N4

f.p. store = M0 M1 M2 M3 M18 M19

string = string store / subsidiary string store / immediate string

subsidiary string store = string store[index] / string store[index:index]

index = f.p. store 0 1 2 3.....78 79

immediate string = "ascii string" / 'hex-pair string'

I/O list = I/O block / I/O list, I/O block

I/O block = f.p. store / string / string*repeat

repeat = f.p. store / number

number = 0 1 2 3 65534 / 65535

ascii string = ascii character / ascii string ascii character

ascii character =	!	"	#	\$	%	&	'	)	(	*	+	,	-	.
	/	0	1	2	3	4	5	6	7	8	9	:	;	<
	@	A	B	C	D	E	F	G	H	I	J	K	L	M
	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\
	^	-	~	{		}	~							

A numerical string (including multipliers such as U, K and MG) can be converted to a floating-point number and stored in a floating-point memory by such statements as:

```
SLET, M1 = N2
SLET, M1 = N2 (Ø:2), "K"
```

Note that the right-hand side must not contain letters other than P,N,U,K MG and G.

Comparison of two strings, and consequent set or reset of the GO/NOGO flags, is obtained by such statements as:

```
SCOMP, N1 ,N2
SCOMP, N2 ,N3(1:4)
```

The length (not the characters) of a string or strings will be logged in the store on the left-hand side of such equations as:

```
SLEN, M1 = N2
SLEN, M1 = N2,N3
```

The total number of characters can be logged as its ASCII equivalent in the statement similar to

```
SLEN, N1 = "THE QUICK BROWN FOX"
```

where N1 will finally contain the ASCII code for 19, which includes spaces in the string.

Note that length runs 1-80, whereas indices run 0-79. It is therefore sometimes necessary, when computing variable indices from length, to subtract 1. Indices must not refer beyond end of string.

Values can be assigned as a string to an N store by such equations as:

```
LET, N1 = 1.Ø
LET, N1 = M1*5
```

In the last example, N1 will store the contents of M1 repeated five times.

A program using the string variable stores could take the form shown below. Explanatory notes are added to the right of some statements

```
S10  LET M2 = 12                      (where 12 is the required voltage)
S20  SU PSU PS1/A TO M2
S30  SLET N1= "SUPPLY IS VOLTS"
S40  SLET N3= 'ØA ØD', "DO YOU WISH TO TRIM?"
S50  APP PSU PS1/A
S55  LET M3 = M2
```


A numerical string (including multipliers such as U, K and MG) can be converted to a floating-point number and stored in a floating-point memory by such statements as:

```
SLET, M1 = N2
SLET, M1 = N2 (Ø:2), "K"
```

Note that the right-hand side must not contain letters other than P,N,U,K MG and G.

Comparison of two strings, and consequent set or reset of the GO/NOGO flags, is obtained by such statements as:

```
SCOMP, N1 ,N2
SCOMP, N2 ,N3(1:4)
```

The length (not the characters) of a string or strings will be logged in the store on the left-hand side of such equations as:

```
SLEN, M1 = N2
SLEN, M1 = N2,N3
```

The total number of characters can be logged as its ASCII equivalent in the statement similar to

```
SLEN, N1 = "THE QUICK BROWN FOX"
```

where N1 will finally contain the ASCII code for 19, which includes spaces in the string.

Note that length runs 1-80, whereas indices run 0-79. It is therefore sometimes necessary, when computing variable indices from length, to subtract 1. Indices must not refer beyond end of string.

Values can be assigned as a string to an N store by such equations as:

```
LET, N1 = 1.Ø
LET, N1 = M1*5
```

In the last example, N1 will store the contents of M1 repeated five times.

A program using the string variable stores could take the form shown below. Explanatory notes are added to the right of some statements

```
S10  LET M2 = 12                      (where 12 is the required voltage)
S20  SU PSU PS1/A TO M2
S30  SLET N1= "SUPPLY IS VOLTS"
S40  SLET N3= 'ØA ØD', "DO YOU WISH TO TRIM?"
S50  APP PSU PS1/A
S55  LET M3 = M2
```

S60 MEAS DCV PIN 7/A9 INTO M1
S70 OUT P1, N1(0:9),M1,N1(9:14),N3
S80 IN, P1 NØ
S90 SCOMP NØ = "YES"
S100 GOIF,GO, S130
S110 SCOMP NØ = "Y"
S120 GOIF,NOGO,S160
S130 LET M3 = M3 -(M1-M2)
S140 SU PSU PS1/A TO M3V
S150 GO TO S60
S160



Rest of program

Output to VDU [see below]

Request reply

Compare keyboard entry

with data stored

Y or Yes if so, go to trim routine

Not required - move on

Simple trim routine

VDU display, from S70:

SUPPLY IS 12.36 VOLTS

DO YOU WISH TO TRIM?

>

keyboard entry

9.9.22 Using the clock

The clock is used as an ASCII string source, which is accessed via four keywords: CLOCK, DATE, TIME, SECS. These may be used in the following statement types:

```
OUT      P1      CLOCK
SLET     N1      = DATE
SCOMP    N2,      TIME
IEEE     OUT,     SECS
PRINT    W10,     DATE, TIME
```

All four keywords yield fixed length strings.

a CLOCK yields a 17-character string composed of the date and time, for example:

```
26/12/82 _ 12:15 _ 39
```

where _ indicates a space character.

NOTE: In a 60Hz machine, the date will be output in U.S.A. type format, e.g.

```
12-26-82 _ 12:15 _ 39
```

For single-digit hours or minutes, the leading zero need not be entered.

b DATE provides an 8-character string comprised of only the date as in the CLOCK output.

c TIME yields a 5-character string comprised of the time, hours and minutes only.

d SECS yields a 2-character string being the seconds readout of the clock.

If the clock has not been previously set up, or the battery discharged destroying the data, the first program access to the clock will cause the message:

```
CLOCK NOT SET
```

to be output to VDU and Printer.

The clock will then be automatically set to:

```
0/ 0/00 00:00 00
```

This allows the rest of the program to run and also allows the clock to be used as a real-time interval timer.

At POWER ON, if a set up CCM is installed, the CLOCK format string will be output to the VDU. Any successive depression of the LOAD/CAL switch whilst in READY mode will also yield the date and time to the VDU.

Further information on the Clock/Calendar module is in Appendix A12.

9.9.23 Syntax Summary

val = fpno or store

fpno = floating point number and optional scaling factor

A number displayed will be a maximum of four digits plus scaling factor

A number input from the keyboard will round off after seven digits plus scaling factor.

Permissible scaling factors are

- P pico
- N nano
- U micro
- M milli
- K kilo
- MG mega
- G giga

Alternative format is:

(decimal number) E (integer)

e.g. 12.34E-3 = 0.01234

Effectively the E integer is a scaling factor and is $\times 10^{\text{integer}}$.

See also section 9.0.1

store = Mx , x = 0-19 (M99=string store)

ival = integer or store

integer may be a max of 65536

Particular limits are context dependent

Range is an integer, 1-8, used to specify voltage ranges:-

1	0.5V	5	8.0V
2	1.0V	6	16.0V
3	2.0V	7	32.0V
4	4.0V	8	64.0V

Omitting a range will cause autoranging or a default range

pinrefs= e.g.56/A5,6/C15 (single)

= e.g.124/A2:C5 (differential)

pol = POS or NEG (P or N in certain instances)

ref = literal component identity. 1 to 6 characters inside " ".

flag = PASS, FAIL, GO, NOGO, NOGOL0, NOGOHI,F0-F7,F99

9.9.24 Test Times

In order to calculate the time taken by any program run, the following table may be used to allocate the test statement duration. The times stated in the table relate to the NORMAL operation of an A8315 and for all operations on A8000.

For details of FAST MODE, see section 9.9.13.

(SAVES 10-15ms!)

Analog Tests - Typical Test Times

<u>Statement</u>	<u>Time</u>	<u>Comments</u>
MEAS DCV/DCI	150ms	20ms Integration Time/Full Scale I/P
MEAS ACV	320ms	200Hz ACM/20ms I.T./Full Scale I/P
MEAS FREQ	100ms	fin = 1kHz
MEAS PER	82ms	fin = 1kHz
MEAS PW	82ms	fin = 1kHz
MEAS EV	112ms + gate time	
PK TR	280ms + (12 x gate time)	

Note: a VER has approximately an extra 3ms overhead provided test PASSES.
ADJ takes as long as the operator is in control + 1 VER.

Stimuli

<u>Statement</u>	<u>Time</u>	<u>Comments</u>
SU PSU	750ms	PCM
SU PSU	22ms	PSM
SU DC	25ms	VIM
SU PULSE	30ms	PGM is stopped
SU SINE	170ms	SSM set to 1kHz

APP any stimulus approximately 15ms.

REM any stimulus approximately 15ms (Note PGM is stopped).

Relays

OP and CL any relay takes approximately 15ms, approximately extra 0.5ms per extra pin.

Digital

<u>Statement</u>	<u>Time</u>	<u>Comments</u>
SET # NAME = XXXX,VER	15ms	
SET CLK 7/A1	12ms	Clock P Width = 11µs
SU SENSE	20ms	Dual Threshold

In-circuit

<u>Statement</u>	<u>Time</u>	<u>Comments</u>
VER R	280ms	
VER C	400ms	200Hz meas freq/200Hz ACM
VER D	650ms	
VER NPN	1.5 s	
Note VER Z = VER C + SU SINE		
VER L = VER C		
GRD	13ms	
TRI	13ms	

Timer

<u>Statement</u>	<u>Time</u>	<u>Comments</u>
CON LTM	13ms	
CON TIM	20ms	
PREP TIMER	16ms	
MEAS TIMER	25ms	Time period has finished

General

All statements have an overhead of 10-15ms*,the bulk of which is removed by MODE FAST (A8315 ONLY).

Mathematical calculations (LET) take approximately 15-20ms.

I/O statements basically depend upon the speed of the I/O device.

EXecute gives a 12ms overhead to the statement being executed; this overhead is present only the first time if there is an iteration parameter greater than 1.

System control statements i.e. Flags, Routeing etc., take approximately 10-15ms.

ZTM Continuity (A8315 ONLY)

Approximately 100 points/sec with defaulted parameters i.e. R = 10Ω

C = 0F.

* Included in the times quoted above.

10.0 MESSAGES & ERRORS

Some of the more common operator messages have been explained in section 3.1 - Operating Instructions. These are included in the comprehensive listing which follows. The system has extensive error detection capabilities, enabling operators to determine the cause and location of an error, and providing information for them to effect a recovery.

The text is in two parts:

- 10.1 Operator Messages and Application Errors
- 10.2 System Errors

10.1 OPERATOR MESSAGES & APPLICATION ERRORS

Collectively, these can be classified into six types:

- a) Statements of the system's current mode (e.g. CALIBRATING). These do not require any operator action, since on completion of the current process the system moves automatically to the next stage of the programming or testing sequence.
- b) Instructions (e.g. INSERT CAL PLATE). The system expects the operator to carry out the instruction and then to press PROCEED.
- c) Information (e.g. TP 0047B2 PASS).
- d) Invitation for input of an optional facility (e.g. DELAY ?). The ? indicates that the programming sequence can be continued, if the facility offered is not required, by pressing <ret>. In this instance, the system adopts a default value. Where an input is made for the optional facility, this is followed by pressing <ret>.
- e) Demand for a mandatory input (e.g. SENSE PIN !). The ! indicates that the programming sequence cannot be continued until the information demanded is input (i.e. there is no default value available).
- f) Advice of an operator error preventing the progress of the programming or testing sequence. This may be where information input is incomplete, invalid or out of sequence; it may be incorrect modules fitted for the program to be run, or it could be caused by test modules not properly connected to the system, or by several other operator failures.

For ease of reference, operator messages and application errors have been combined into a single alphabetical list. (Messages commencing with symbols are shown at the end). Many of these are self-explanatory: for the rest, either information is provided or a reference is given to the appropriate section of this Manual where clarification can be found.

Operator Messages and Application Errors

(OM) = Operator Message

! = Mandatory input

(AE) = Application Error

? = Optional input

(/D) = Disc

AA,AB,BA OR BB EXPECTED	(AE)	TPG - only modes AA,AB,BA or BB are allowed. (Early A8000's only).
A OR K EXPECTED	(AE)	TPG - neither A nor K specified, A = Anode, K = Cathode.
ANODE PIN!	(OM)	IPG prompt for a ZTM pin reference.
AUTO LINE NUMBER / INC =	(OM)	TPG X/Y <ret> line start line X incrementing by Y lines (e.g. 5/6 = start line 5, increment by 6 lines). <ret> (default) = 10/10.
<u>A8315 "COMBAT" SYSTEM</u>	(OM)	Control package.
<u>A8000 AUTOMATIC TEST SYSTEM</u>	(OM)	Control package.
BANDWIDTH (EB = EXTENDED) ?	(OM)	IPG prompt for the selection of Extended Bandwidth. EB should be typed if required, else <ret>.
BANK LETTER ERROR	(AE)	TPG Module data correct, but illegal bank data given (e.g. 7/B2 for LTM).
BASE !	(OM)	IPG prompt for a ZTM pin reference.
BIT PATTERN TOO LONG	(AE)	TPG - this bit pattern would require more digital pins than have been declared in the DEF NAME.
BUFFER OVERFLOW	(AE)	TPG While setting clock more than 128 characters have been input.
CALIBRATING	(OM)	TPG Indication that the system is in the process of calibration. This may be the result of pressing the CAL button on the console or by typing C at the FUNCTION level of the FUNCTIONAL PROGRAM GENERATOR.
CATHODE PIN !	(OM)	IPG prompt for a ZTM pin reference.
CCM NOT FITTED	(OM)	Interpreter - hardware not fitted, statement aborted.
CLOCK NEG TIMES	(AE)	Interpreter - neg. no. of times irrational.

CLOCK NOT AVAILABLE	(AE)	Interpreter - software not fitted.
CLOCK NOT SET	(OM)	Interpreter. Clock was not set up but has now been started at 00/00/00 00:00 to allow timing measurements.
CLOCK SET TO	(OM)	TPG The date and time is now set to the following.
CODE NOT SAVED IN EPROM	(OM)	TPG - program has not successfully been stored on EPROM. (A8000 only).
COLLECTOR !	(OM)	IPG prompt for a ZTM pin reference.
COMMENT ?	(OM)	IPG prompt for a comment. This will be lost if not printed when entered.
COMPONENT REFERENCE !	(OM)	IPG prompt for a component reference which may be up to six characters long. Any ASCII character acceptable including spaces.
CONNECT ONTO !	(OM)	IPG prompt for CTA or CTB.
CONTINUITY GENERATOR VO1		Implies SOM continuity generator selected.
COPY DISC FUNCTION	(OM/D)	TPG Indication that 'FC' has been selected at FUNCTION level in the Functional or Continuity Generator.
CTA OR CTB EXPECTED	(AE)	TPG Control line Counter Timer A or B required by program.
CTA,CTB OR GTO EXPECTED	(AE)	TPG Control line Counter Timer A or B or GOTO required by program.
CURRENT LIMIT ?	(OM)	IPG PSM current limit prompt.
CYCLES (16, 256)?	(OM)	IPG prompt to force number of cycles used for measurement of frequency.
DELAY ?	(OM)	IPG prompt for delay length (seconds).
DELETE PROGRAM FROM DISC	(OM/D)	TPG Indication that 'FD' has been selected at FUNCTION level in the Functional or Continuity Generator.
DELETE LEARNT CONTINUITY PROGRAM (Y OR N) ?	(OM)	TPG Confirmation required for action requested (A8315 only).
DIAGBEGIN MISSING	(OM)	Interpreter - A.D.D.
DIAGEND MISSING	(OM)	Interpreter - A.D.D.
DISC ERROR	(OM/D)	Code 10, Track 0 not found. Code 11, Write fault.
DISC FULL	(OM/D)	

DISC NOT AVAILABLE	(OM/D)	TPG - disc system software absent.
DISC PROGRAM NAME ?	(OM/D)	TPG prompt for entry of a disc program name.
DRIVE ANODE (A) OR CATHODE (K) !	(OM)	IPG prompt for which end of a diode to drive during an in-circuit test.
DRIVE NOT READY	(OM/D)	Disk missing or door open.
DRIVE PIN !	(OM)	IPG prompt for a ZTM pin reference.
DRIVE PIN = SENSE PIN	(AE)	Interpreter. Only one terminal of a component may be specified as being connected to the same ZTM pin.
DUPLICATE LINE NUMBER	(AE)	TPG When editing a functional test statement, the statement number has been altered to a number which already exists. <ret> will overwrite previous statement with new statement.
EARLIER SRQ MISSED	(OM)	Nucleus - service request (SRQ) was flagged on IEEE bus but withdrawn before being serviced.
EMITTER !	(OM)	IPG prompt for a ZTM pin reference.
EOS ?	(OM)	IPG prompt for end of sequence character.
EPROM NAME MISMATCH	(AE)	Duplicate message, repeat RELOAD CORRECT JIG message.
EPROMS NOT EMPTY	(AE)	Continuity generator. (A8000 only).
EPROM NOT FULLY ERASED	(OM)	TPG. (A8000 only).
EPROM READ FAIL	(AE)	Interpreter.
EPROM SET CAN BE OVERWRITTEN	(OM)	TPG. Options are

- 1 Abort

2 Overwrite

3 Use new set

}

A8000 only

EPROM SET CANNOT BE OVERWRITTEN (OM) TPG. One of four groups of options will be shown:

- A : 1-Abort/2-Use new set

B : 1-Abort/2-Use a new set/
3-Reload correct jig

}

A8000 only

C : 1-Abort/2-Insert additional PROMS/

3-Use a new set

D : 1-Abort/2-Replace faulty EPROM and retry/

3-Use a new set

A8000 only

EPROM UPDATED CORRECTLY	(OM)	TPG (A8000 only)
EPROM WRITE FAILURE	(OM)	TPG Written program does not match RAM program. (A8000 only)
ERR PIN x/by	(AE)	(ATM pin not calibrated, section 3).
EXCESS FWD BASE I	(OM)	Interpreter - Base current excessive, possible S/C.
EXCESS REV BASE I	(OM)	
EXECUTE/GOTO ERROR	(AE)	Interpreter. System will not execute a GOTO or GOIF from an executable statement or to an unspecified statement.
EXECUTE JUMP ERROR	(AE)	
EXIT	(OM)	TPG Control has been passed back to the test console from the Functional or Continuity program Generator.
EXTRA DIAGBEGIN	(OM)	Interpreter - A.D.D.
EXTRA DIAGEND	(OM)	Interpreter - A.D.D.
EXTRANEIOUS CHARS AFTER LAST PARAMETER	(AE)	TPG Last parameter is valid but is followed by unrecognised characters.
FAULTY	(AE)	Interpreter - Diode or Transistor diagnostic.
FLAG REF TOO LARGE	(AE)	Interpreter The test flag specified is not F0-F7 nor F99.
FLAG TO BE RESET (F0-F7,F99) !	(OM)	IPG prompt for a test flag which may be F0-F7 or F99.
FLAG TO BE SET ! (PASS,FAIL,GO,NOGO,NOGOHI, NOGOLO,F0-F7,F99)	(OM)	IPG
FREQUENCY !	(OM)	IPG
FROM (CTA,CTB OR GTO)?	(OM)	IPG
FUNCTION ?	(OM)	TPG Command level. See Section 8.1.
FWD OR REV EXPECTED	(AE)	TPG Transistor partial test following ONLY.
GATE TIME ?	(OM)	IPG prompt for measurement time for PK, TR or Events. Defaults to 20ms.

GUIDED PROBE NOT AVAILABLE	(OM)	TPG A.D.D. option not fitted.
HIGH THRESHOLD !	(OM)	IPG
HI LIMIT?	(OM)	IPG limit for verification of test.
IEEE CARD NOT FITTED	(OM)	Nucleus
IEEE COMMAND !	(OM)	IPG prompt for an IEEE command, i.e. DCL,SDC,GET,IFC,LL0,GTL or LOC.
ILLEGAL EXECUTE	(AE)	Interpreter. System will not execute a GOTO or GOIF from an executable statement or to an unspecified statement.
ILLEGAL/MISSING PARAMETER	(AE)	TPG - statement not constructed correctly, possibly a missing comma.
ILLEGAL MODE	(AE)	Interpreter - mode unrecognised.
ILLEGAL MODULE IDENTIFIER	(AE)	TPG - not a recognised module type.
ILLEGAL TR MODULE	(AE)	Continuity generator. Functional module found in Test Rack.
INCOMPATIBLE ISSUE	(AE)	Interpreter. Current software will not support this function.
INCOMPATIBLE SOFTWARE	(AE)	TPG Current software will not allow this function. Retry statement.
INCORRECT FUNCTION LETTER	(OM)	TPG See section 8.1. No command matches letters typed.
INITIALISE DISC ?	(OM/D)	TPG Answer Y destroys disk contents.
INITIALISING	(OM/D)	TPG - disc is being initialised.
INPUT YEAR ?	(OM)	TPG Prompts for clock setting. Enter year in 2 or 4 digit format. If clock is already set, these may be defaulted.
INPUT MONTH (1-12) ?		
INPUT DATE (1-31) ?		
INPUT TIME (HRS:MINS) ?		
INSERT CAL PLATE	(OM)	Insert calibration plate, then press PROCEED, or exit by pressing RESET.
INSUFFICIENT EPROMS	(AE)	Continuity generator. (A8000 only).
INTEGRATION TIME ?	(OM)	IPG Defaults to 20ms.
INTO STORE ?	(OM)	IPG Prompt for M0 - M19.
INVALID	(AE)	TPG Clock set-up data out of range or wrong format.
INVALID BASE	(AE)	TPG Digital: specify Binary, Octal, Decimal; by default - Hexadecimal.

INVALID COMMAND	(AE)	TPG
INVALID FLAG SYMBOL	(AE)	TPG Attempt to SET/UNSET INVALID flag.
INVALID FLOATING POINT NUMBER	(AE)	TPG Format not recognised.
INVALID LET EXPRESSION	(AE)	TPG - LET statement not constructed correctly.
INVALID MODE	(AE)	TPG
INVALID MODULE NUMBER	(AE)	TPG
INVALID NUMBER	(AE)	TPG <u>or</u> (AE) Interpreter.
INVALID OP CODE	(AE)	TPG Data has been corrupted, memory or media.
INVALID PIN NUMBER	(AE)	TPG - pin number not allowed for this type of module.
INVALID POLARITY SYMBOL	(AE)	TPG - not POS,NEG,P or N.
INVALID START IGNORE	(AE)	Interpreter. PGM, continuous pulse train already running.
INVALID STOP IGNORE	(AE)	Interpreter. PGM already stopped.
INVALID STRING DELIMITER	(AE)	TPG missing " or '.
INVALID UNITS	(AE)	TPG - units incorrect for this parameter.
I/O LIST !	(OM)	IPG Refer to string syntax.
I/O LIST ?	(OM)	IPG
IPG V01 IPG V02	(OM)	TPG Indication that the Interactive Program Generator has been entered as a result of typing 'I' at the FUNCTION level of the Functional Program Generator.
ITERATIONS ?	(OM)	IPG prompt for the number of times the operation is to be repeated.
JIG FAILURE	(AE)	TPG Program not registered via PLM. (Some A8000s only).
JIG UPDATED CORRECTLY	(OM)	TPG The system has successfully stored the test program on EPROM. (Some A8000s only).
LENGTH,BASE ! <BASE OPTIONAL>	(OM)	IPG prompt for the digital name attributes. See 9.7.3.
LIMIT NOT LOADED	(AE)	Interpreter - memory referred to has not been assigned a value.

LIN FWD COLLECTOR I LIN REV EMITTER I	(OM)	Interpreter - measured current is not non-linear
LINE DELETED/UPDATED	(OM)	TPG - previously stored statement now altered.
LINE NUMBER <1, or >9999	(AE)	TPG Program characterisation. Statement numbers must be in the range 1 to 9999.
LIST DISC DIRECTORY	(OM/D)	Indication that 'FL' has been selected at FUNCTION level in the Functional or Continuity Program Generator.
LIST LINES	(OM)	Continuity generator. Indication that 'L' has been selected at FUNCTION level of SOM Continuity Program Generator and a listing of test program statements will follow on the screen.
LIST LINES FROM/TO:	(OM)	TPG Program characterisation. Indication that 'L' has been selected at FUNCTION level of Functional Program Generator and system is waiting for input from keyboard to select which statements are to be listed on the screen with optional continuity listing. See section 9.6.3.
LISTENER(S) !	(OM)	TPG (IEEE).
LOADING	(OM)	System is loading TPM program.
LO LIMIT ? LO LIMIT !	(OM)	IPG prompt for a floating point number to be used as the low limit for the test.
LO THRESHOLD !	(OM)	IPG
MANUAL LINE NUMBERING	(OM)	TPG Program characterisation. Indication that 'M' has been selected at the FUNCTION level of the Functional Program Generator and that the generator is no longer in Auto-line numbering mode.
MARK TIME !	(OM)	IPG
MARK VOLTS ?	(OM)	IPG
MAX CAPACITANCE ?	(OM)	IPG Maximum board capacitance.
MAX DRIVE VOLTS ?	(OM)	IPG Maximum voltage at which system will seek diode conduction.

MEM REF TOO LARGE	(AE)	Interpreter - reference to memory which is not M0-M19, N0-N4 nor M99.
MEM VALUE TOO LARGE	(AE)	Interpreter
MK/SP RATIO EXCEEDED	(AE)	Interpreter PGM statement.
MISSING /	(AE)	TPG / missing.
MISSING :	(AE)	TPG : missing.
MOD. NO !	(OM)	IPG prompt for Module number.
MODULE NO. ALLOCATED	(AE)	TPG - module number has already been allocated in a previous Declarative Statement.
MODULE SOFTWARE MISSING	(AE)	TPG No software found to match a test rack module.
MODULE S/W MISSING	(AE)	Control package. No software found to match a test rack module.
MODULE UNDEFINED	(AE)	TPG The module number has not been declared in a Declarative Statement.
MODULE 4 NOT A SSM	(AE)	Interpreter - this in-circuit test requires module 4 to be a Sinewave Source Module.
NAME !	(OM)	IPG prompt for the input of a new or previously defined digital name.
NAME ALREADY DECLARED	(AE)	TPG - a DEF NAME statement already exists for this Digital Name.
NAME LENGTH ERROR	(AE)	TPG Digital name in excess of 6 characters.
NAME NOT DECLARED	(AE)	TPG. No DEF NAME statement for this Digital Name.
NEG. CURRENT LIMIT ?	(OM)	IPG prompt for a negative current limit. VIM.
NEG PULSES	(AE)	Interpreter - negative number of pulses is irrational.
NEITHER HL OR LL SPECIFIED	(AE)	TPG - either High limit, Low limit or both limits must be specified.
NEW DISC NAME ?	(OM/D)	TPG disc initialisation. Assign a name to the disc being initialised. The name may be up to 8 characters long. Escape with CTRL Q.

NEW EPROM SET TO BE USED ?	(OM)	Maximum program size reached. It may be possible to compact the program if a new set of EPROMs are used. If this is desired type 'Y'. (A8000 only).
NEW (N), EXISTING (E) OR DISC (D)	(OM)	TPG SOM continuity programming invocation.
NEW (N), EXISTING (E), DISC (D) OR RECEIVE (R)	(OM)	TPG Functional programming invocation.
NO ACM	(AE)	Control package. Probable connection fault.
NO CHANGE RECORDED	(OM)	TPG
NO CONSOLE	(OM)	Control package. Indicates Test Console test pins o/c or hung due to poorly connected module(s).
NO DRIVE	(OM/D)	Disc hardware not found.
NO EXISTING PROGRAM	(AE)	TPG Program not registered via PLM.
NO EXTENSION DRIVE	(OM/D)	No drive fitted.
NO FWD COLLECTOR I	(OM)	Interpreter. Collector current did not reach 1mA.
NO IEEE DEVICE	(OM)	Nucleus. There are no IEEE devices on the Bus.
NO OF COLS (5,10,20,40) !	(OM)	IPG prompt for the maximum number of characters to be printed on one line of the ticket printer.
NO PCM - PARTIAL CAL	(OM)	Calibration package.
NO PLM	(AE)	Control package. Probable connection fault.
NO PREVIOUS TRIGGER	(AE)	Interpreter, PGM. Previous START command was not triggered.
NO PROGRAM	(OM)	Control package. Connect TPM with program and press LOAD/CAL.
NO PULSES - TIME OUT	(AE)	Interpreter. Measurement of event, frequency, pulse width etc. incomplete :FØ set.
NO REV EMITTER I	(OM)	Interpreter. Reverse drive emitter current did not reach 1mA.
NO SOM IN TEST RACK	(OM)	Continuity generator. (SOM).

NO SPACE LEFT FOR PROGRAM	(AE)	TPG Maximum program size reached with A8000, press return and retry, EPROM overwrite capability then lost.
NO ZTM IN TEST RACK	(AE)	Interpreter. COMBAT only, attempt to run continuity statement with no ZTMs.
NO. OF PULSES ?	(OM)	IPG Default gives continuous pulses.
NOISE IMMUNITY ?	(OM)	IPG TIM Hysteresis.
NOMINAL VALUE !	(OM)	IPG Centre point for ADJust display.
NON-BINARY DIGIT (e.g. 3)	(AE)	TPG On assigning a digital pattern to a digital name in a SET Statement, the digital pattern contains a digit which is not of the same base as specified for that name in its DEF NAME statement.
NON-DECIMAL DIGIT (e.g. B)		
NON-HEX DIGIT (e.g. J)		
NON-OCTAL DIGIT (e.g. 8)		
NOT A CONTINUITY PROGRAM	(AE)	SOM Continuity generator. An attempt has been made to load a program from storage media into the SOM Continuity Program Generator when it is not an SOM continuity program.
NOT A FUNCTIONAL PROGRAM	(AE)	TPG An attempt has been made to load a program from storage media into the Functional Program Generator when it is an SOM continuity program.
NOT FOUND	(OM/D)	123ABC (prog name) not found on disc (see section 5.1.).
NUMBER OF PINS INCORRECT	(AE)	TPG When constructing a DEF NAME statement the number of digital pins listed is different from that declared in the Length parameter.
NUMBER OUT OF RANGE	(AE)	TPG - the value entered for the parameter is outside the parameter's specification.
OPEN CIRCUIT PAIRS	(OM)	Interpreter. Continuity failure header.
O/C OR RESISTIVE	(AE)	Interpreter, diode or transistor diagnostic.
OUT NEG ITERATIONS	(AE)	Interpreter (Cannot output anything a -ve number of times).
OUTPUT LINES TO PORT 2	(OM)	SOM Continuity generator. Indication that 'O' has been selected at the FUNCTION level of the Continuity Program Generator and that the program 'listing' is being sent out of SIO port 2 (if fitted).

OUTPUT (PORT 2) FROM/TO:	(OM)	Functional Generator. Program listing is being output by AUXILIARY SIO port. Indication that 'O' has been selected at the FUNCTION level of the Functional Program Generator and the system is waiting for input from the keyboard to select which statements are to be sent out of SIO port 2 (if fitted). See "O" on page 8-9.
OUTPUT VOLTAGE ?	(OM)	IPG
OVERRANGE ERROR	(AE)	Interpreter. Measured voltage too great for range specified.
OVERWRITE THIS DISC ?	(OM/D)	A double check that the user wishes to overwrite the disc whose directory is on the screen. Typing 'Y' will overwrite, anything else will abort.
PARTIAL TEST (FWD OR REV) ?	(OM)	IPG
PATTERN/PIN CHANGE !	(OM)	IPG prompt for a digital pattern or pin change information. See symbolic digital programming 9.7.4.
PERM ERROR	(OM/D)	Disc error message (PERM = permanent), after several retries.
PIN !	(OM)	IPG prompt for a pin reference. Mod no/bank, pin.
PIN REQD MEAS	(OM)	Interpreter. Header for digital failure list.
PIN REFS ! PIN REFS ?	(OM)	IPG prompt for one or more pin references.
POLARITY (P OR N) ?	(OM)	IPG prompt for the input of the polarity parameter, P = Positive, N = Negative. Default generally negative.
PORT NO !	(OM)	IPG prompt for an SIO port number 1 = VDU, 2 = 2nd SIO.
POS. CURRENT LIMIT ?	(OM)	IPG prompt for a positive current limit.
PRESS PROCEED WHEN READY	(OM)	TPG
PREVIOUS SET UP INVALID	(AE)	TPG. Clock is not set up and so set up input data may not be defaulted.
PRINT DISC DIRECTORY	(OM/D)	Indication that 'FP' has been selected at FUNCTION level in the Functional or SOM Continuity Program Generator.

PRINT LINES	(OM)	SOM Continuity Generator. Indication that 'P' has been selected at the FUNCTION level of the Continuity Program Generator and that the program 'listing' is being printed on the ticket printer (if switched on).
PRINT LINES FROM/TO:	(OM)	TPG Indication that 'P' has been selected at the FUNCTION level of the Functional Program Generator and the system is waiting for input from the keyboard to select which statements are to be printed on the ticket printer (if switched on). See "P" on page 8-9.
PROGRAM CONSTRUCTED	(OM)	Indication that SOM Continuity Program Generator has completed the self-learning of the test program for the UUT.
PROGRAM DELETED	(OM/D)	TPG deleted program from Disc.
PROGRAM FORMAT INCORRECT	(OM)	Attempting to receive a functional test program from an external device through SIO port 2, the characters received do not match the specification described in section 8.1.
PROGRAM IS	(OM)	TPG Giving program name.
PROGRAM NAME (6 CHAR: 0-9,A-F)?	(OM)	Requesting program name. Either NEW has been selected and the system waits for the name of the program which is about to be created. Or NN has been selected and the system waits for the new name for the program already created. Escape with CTRL Q.
PROG NAME MISMATCH	(AE)	Control package. The test program being run no longer has the same name as selected on the TPM.
PROGRAM TOO LARGE	(AE)	TPG
PROGRAMMED LEVEL !	(OM)	IPG
PROGRAMMED VALUE ! PROGRAMMED VALUE ?	(OM)	IPG In Digital Threshold, defaulting forces dual threshold prompts.
PROGRAMMING	(OM)	Indication that control has been passed from the test console to the keyboard.
PSU NO !	(OM)	IPG Module No/Pin Bank.
RANGE (1-8)?	(OM)	IPG prompt to force a voltage Range. See section 9.0.3.

RE-ENTER GENERATOR (Y OR N)	(OM)	Prompt to use previous program.
RE-NAME PROGRAM	(OM)	TPG Indication that 'NN' has been selected at the FUNCTION level of the Functional or SOM Continuity Program Generator, and the operator is able to assign a different name to his test program.
RE-NUMBERING	(OM)	TPG Indication that 'NL' has been selected at FUNCTION level of the Functional Program Generator and that the system is re-numbering the test statements.
READY	(OM)	The test console is waiting for a command.
RELOAD CORRECT JIG	(AE)	TPG The edited EPROM program does not have the same name as that on the TPM. (Applicable to some A8000's only).
REMOVE CAL PLATE	(OM)	Calibration package. Remove calibration plate and press 'Proceed'.
REQUESTED ATM UNCAL	(OM)	Operator has pressed RESET to bypass ATM Calibration. Press PROCEED to acknowledge message.
RESOLUTION ERROR	(AE)	Interpreter The high limit, low limit and nominal for an adjust statement cannot all be displayed on the segments, e.g. low limit greater than high limit.
REVERSED	(AE)	Interpreter, diode diagnostic.
RUN PROGRAM FROM RAM	(OM)	Indication that 'R' has been selected at the FUNCTION level of the Functional or SOM Continuity Program Generator and that the system is executing the program presently resident in that generator.
RUNNING FAST MODE	(OM)	Interpreter. Display message while Fast Mode is operative.
RUNNING MODE (FAST,NORMAL,UNCAL) !	(OM)	IPG. Type in full the first or only Mode required
RUNNING MODE ?	(OM)	IPG. Type in full the second mode required or CR to default.
SAVE PROGRAM ?	(OM)	TPG This is a warning to the user that if the test program currently being created/edited is not stored on storage media, it may be lost. Typing 'N' will continue, anything else will abort.
S/C OR RESISTIVE	(AE)	Interpreter, diode or transistor diagnostic.

SECTOR NOT FOUND	(OM/D)	Disc data corrupted.
SEEK FAIL	(OM/D)	Seek track # command failure.
SELECT	(OM)	IPG, Menu prompt.
SELECT CONTINUITY (C) OR FUNCTIONAL (F)	(OM)	The program button has been pressed on the console and a choice needs to be made on what type of program is to be created or edited.
SELF LEARNING	(OM)	The SOM Continuity Program Generator is in the process of generating test statements for the UUT.
SENSE PIN !	(OM)	IPG prompt for a ZTM pin reference.
SHORT CIRCUIT PAIRS	(OM)	Interpreter, Continuity failure header.
SHUNT RESISTANCE !	(OM)	IPG DC current requirement.
SIGON MISSING	(OM)	Interpreter. A.D.D.
SPACE TIME !	(OM)	IPG
SPACE VOLTS ?	(OM)	IPG
START POLARITY !	(OM)	IPG
STATEMENT NO !	(OM)	IPG prompt for the statement number for the statement about to be created.
STATEMENT TOO LARGE	(AE)	TPG too many characters in statement.
STMT NO !	(OM)	IPG prompt for a statement number to be used in an EX,GOTO or GOIF control statement.
STOP POLARITY !	(OM)	IPG
STORE NOT LOADED	(AE)	INT An attempt has been made to use a memory (M0-M19) which has not been assigned a value.
STRING !	(OM)	IPG
STRING INDEX ERROR	(AE)	INT Index printer greater than string length (minus one).
STRING LENGTH ERROR	(AE)	TPG The string is either of zero length or is greater than allowed for this application.
STRING STORE ! STRING STORE ?	(OM)	IPG prompt for a string store (N0-N4).
STRING STORE EXPECTED	(AE)	TPG Expects N0-N4.

SUPPLY MAXIMUM CIRCUIT CAPACITANCE	(OM)	SOM Continuity generator. Prompt may be defaulted.
SUPPLY OPERATOR INSTRUCTION (RET IF NONE)	(OM)	SOM Continuity generator.
SUPPLY SHORT/OPEN THRESHOLD RESISTANCE (5-50 OHMS)	(OM)	SOM Continuity generator. Prompt may be defaulted.
TALKER !	(OM)	IPG
TALKER(S) ?	(OM)	IPG
TEST FLAG ? (PASS,FAIL,GO,NOGO,NOGOHI, NOGOLO,F0-F7,F99)	(OM)	IPG
TEST RACK MISMATCH e.g. TEST RACK MISMATCH PRESENT IN RACK MOD ADDR ZTM 16	(AE)	Control package. Turn printer on and press Load/Cal. Test Module Addresses in Rack do not match declaratives in Program.
REQUIRED BY PROGRAM MOD ADDR SLOT ZTM 16 SSM 4 6		Therefore, SSM missing from Test Rack.
THRESHOLD LEVEL ?	(OM)	IPG TIM input threshold for timing.
TIME !	(OM)	IPG prompt for the length of time to delay in seconds.
TIME OUT ?	(OM)	IPG Default infinity
TIMER MODE !	(OM)	IPG prompt for timer mode.
TIMER IN USE	(AE)	Interpreter. PREP TIMER still awaits MEAS or VER,timer is used for all analog measurements.
TIMER NOT PREPARED	(AE)	Interpreter. MEAS or VER TIMER not preceded by PREP TIMER.
TOO MANY PINS * RE-ENTER ALL OF PIN LIST	(AE)	TPG - more pins have been entered than declared in the Length parameter.
TPM NOT CONNECTED	(AE)	TPG
TRANSMIT WITH HANDSHAKE (PORT 2) FROM/TO:	(OM)	TPG Sending program to a second System. Indication that 'T' has been selected at FUNCTION level of the Functional Program Generator and the system is waiting for input from the keyboard to select which statements are to be sent

		out through port 2 using the protocol described in section 8.1.
TYPE OPTION SELECTED	(OM)	TPG A8000 EPROM write facility.
UPDATE EXISTING PROGRAM ?	(OM)	A check with the user that the program already on the disc with the same name is to be overwritten.
UPDATING	(OM)	TPG Stored EPROM program is being updated. (A8000 only).
VARIABLE STORE ! VARIABLE STORE ?	(OM)	IPG prompt for a variable store (M0-M19).
VARIABLE STORE EXPECTED	(AE)	TPG (M0-M19)
VARIABLE STORE = EXPRESSION !	(OM)	IPG prompt for a let expression e.g. M1 = M2*M3+M4.
WRITE ERROR	(AE)	Failure to store program on EPROMS. (A8000 only).
WRITE EPROM FUNCTION	(OM)	Indication that 'W' has been selected at FUNCTION level of the Functional or Continuity Program Generator. (A8000 only).
WRITE EPROM FUNCTION NOT AVAILABLE	(OM)	Indication that 'W' has been selected at FUNCTION level of the Functional or Continuity Program Generator. (A8315 only).
WRITE FAIL	(OM/D)	Verify mismatch - Data written twice.
WRITE PROGRAM TO DISC	(OM/D)	Indication that 'FW' has been selected at FUNCTION level of the Functional or Continuity Program Generator.
WRITE PROTECT	(OM/D)	Writing not possible because disc has been protected.
WRITING	(OM)	TPG Program is being written to storage media.
WRONG PROG STRUCTURE	(AE)	Control package. Poor PLM/TPM connections, faulty or poorly sited TPM EPROMS. Indicates corrupted program.
: OR) EXPECTED	(AE)	TPG : or) missing.
, OR) EXPECTED	(AE)	TPG , or) missing.
) EXPECTED	(AE)	TPG) missing.

10-18

(section 10.1 continued)

= EXPECTED

(AE) TPG = missing.

60HZ MAINS

(OM) Integration time adjusted for 60Hz.

200HZ ACM

(OM) Control package. High speed ACM fitted. LF response down 0.1dB at 200Hz instead of 20Hz.

10.2 SYSTEM ERRORS

A System Error arises when the software finds an illogicality during execution of either the users test program, or the execution of the internal operating system. In the majority of instances it is a warning that there is a hardware fault. It is therefore advisable to contact the Customer Services Department of Wayne Kerr for advice regarding the problem.

In some cases it may be possible to recover by pressing "RESET" and re-running the sequence that gave the System Error, after first ensuring that the correct Test Modules are properly positioned for the particular application and making sure that they are firmly located in their relevant Test Interface Connectors. If the system error occurred whilst running a program from TPGs, it is possible to recover by pressing RESET and then pressing PROGRAMMING. The user can then select C or F depending upon whether the error occurred in continuity mode or functional mode. The system may then ask if re-entry is required. An answer of Y to this question will allow entry to the FUNCTION level without destroying the contents of RAM. An N answer to the question will result in the system clearing RAM and requesting either a new or an existing program.

The format of a System Error is:-

*SYSERR 05:7F:0350 *

The first number (in this case 05) is the actual Error Number. The second number (in this case 7F) is a byte containing information as to the cause of the error (Accumulator Contents). The last number (in this case 0350) is the location at which the error was detected.

Error numbers C2, C5 & C6 plant extra information in the 4 digit section and this is indicated by the @ sign in the following tables.

System Errors

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
NUCLEUS	00			
	01	System device code not recognised	Device code	RAM Failure/Corruption
	02	System device Exception Byte	Device/Command	" " "
	03			
	04	TMH call illogical (BS address not found)	Module Address	" " "
	05	TMH exception on exit (most significant byte only)	Exception Byte	Illogical request from a test module. Especially when using memory-referenced parameters. PROM/RAM.
	06	Unaccepted Interrupt	Interrupt Level	Either (A) A spurious interrupt with no vector or (B) Reading any RST instruction from memory i.e. ROM failure (generally has READ FF as op code).
	07	Invalid Maths Pack Command	(Command-1)*2	ROM/RAM Failure/Corruption
	08	SWPUSH Error		" " " "
	09	SWPOP Error		" " " "
	0A	Block Switch Error		Handler not fitted, or corrupted.
	0B	System entered idle loop (Stack error)		ROM failure.
	0C	Maths function only on H/W chip		Requesting a scientific mathematics function with no high speed maths option.
	0D	Maths Chip Failure		Failure, or no chip.
	0E	Service tool not available		No debugger fitted, or wrong address.
	0F	Interrupt overflow counter full		Overflow on timer measurements i.e. > 34 minutes.
	10	RS232 Port 2 character send fail		Unable to send character to second I/O port.
	11	Disc Handler not found		EPROM not fitted or corrupted/BSM fault.
	12	Clock Handler not found		[As 11]
TPG	13			
	14			
	15	EPROM read failure	PLM status	PEM/PCM/TPM failure. Next byte displayed is PCM hardware status.
	16	Unable to delete found entry in vector table	Status	Usually program failure, i.e. RAM failure or user program failure (PROM or DISC suspect).
	17	Unable to delete found entry in declarative table	Status	
	18	Unable to add declarative in vector table	Status	
	19	Unable to add declarative in declarative table	Status	
	1A	Unable to add location in vector table	Status	
	1B	Unable to decode op-code	Op-code	
	1C	Declaration type code error	Type-code	

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
TPG	1D	Statement length different to decode length	Length difference	Usually program failure, i.e. RAM failure or user program failure (PROM or DISC suspect).
	1E	Statement field of type byte invalid	Type AND E0	
	1F	Floating point number type byte invalid	Type Byte	
	20	Integer type byte invalid	Type Byte	
	21	S/T field or string invalid	Type and F8	
	22	Flag type byte not valid	Type Byte	
	23	Polarity type byte not valid	Type Byte	
	24	Polarity value not + or -	Value	
	25	Invalid operator code in expression	Code	
	26	ZRINT (PLM initialisation) fail		
	27	Floating point number failure	MP status	
	28	ZRINT (PLM initialisation) fail after reading program	PLM status	
	29	IPG Table look-up error		
	2A	Tables cannot be constructed	TPT Status	
	2B	Subcode Block terminator expected	Code	
	2C	Invalid name Type-code	Code	
	2D	Invalid truth-Table type code		
	2E	Mode type byte not 11	Type Byte	
	2F	Counter timer type byte not 11	Type Byte	
CONTINUITY GENERATOR	30	Zero EPROMS requested in preamble		If next byte displayed is FF, no SOM's in rack. Otherwise an SOM cannot be found.
	31	Table construction fault	Status	
	32	Missing handler	Status	
	33	Incorrect op-code for CONTINUITY PROGRAMMING		
	34	No SOM found		
	35			
	36			
	37			
	38			
	39			
TPG	3A			RAM Failure/corruption " " "
	3B			
	3C			
	3D			
TPG	3E	INVALID IEEE COMMAND	Command code	
	3F	INVALID STRING HEADER	Header	

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
INTERPRETER	40	Illegal op-code	Op-code	PROM/Disc read error or RAM failure/corruption.
	41	Statement exceeds 127 bytes		
	42			
	43			
	44			
	45	Delay calculation error, or from Maths Pack (MP)	MP error code	RAM/maths pack error.
	46	Let error from Maths Pack	MP error code	" " " "
	47			
	48	Immediate limit not normalised (wrong format)	MP error code	" " " "
	49	IN CCT Interpreter not found		In-circuit interpreter not fitted, or corrupted, or BSM error.
INTERPRETER (DIGITAL)	4A	Macro handler Maths Pack error		RAM failure/corruption.
	4B	CMRR DAC not acting for single or differential pins	Pin ref type	ACM or DAC failure.
	4C	PCM mod address not 1 or 2 or 3		RAM/PROM Disc error.
	4D	Continuity Interpreter not found		Continuity interpreter not fitted, or corrupted/BSM fault.
	4E	Digital Interpreter not found		Digital interpreter not fitted, or corrupted/BSM fault.
	4F	SOM not found	Mod No.	SOM module not found at run time.
	50	SSM trim error		SSM failure, or frequency count error (TRC failure).
	51	Maths pack error	Mod Address Length	PROM/Disc read errors or out of range parameters when using memory referencing.
	52	Set up sense threshold <0 or >13.5		
	53	Maths Pack error threshold conversion		
	54	EPROM read error		
	55	LTM not found		
INTERPRETER (CONTINUITY)	56	Illegal sub-code (digital operation)		PROM/Disc/RAM error.
	57	Module not in MODSTAB		
	58	Name not declared		
	59	Name too long		
	5A	Illegal statement passed to cont int		PROM/Disc/RAM error.
	5B	Illegal op-code (continuity)		
	5C	Floating point number to ASCII conversion error		
	5D			
	* 5E	Illegal probe pin	Mod address	RAM/PROM/Disc error.
	5F			

* INTERPRETER (DIGITAL)

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
TEST MODULE HANDLERS	60	SOM hardware fault	Mod No	SOM faulty, incorrect threshold on 'grounded' pin. Module No is next byte displayed; or ACM DACOUT faulty.
	61	RTM fault	Command	Either invalid numbers have been passed over or GTO (PGO) not working.
	62	RRM illegal pin	Mod No	Illegal pin in RRM asked for = RAM/storage error.
	63	PCM protocol error	Mod No	PCM protocol flip-flop not working, or decode failure.
	64	ATM maths error	Mod No	RAM failure/corruption.
	65	DAC setting error		SSM tried to set voltage >8V i.e. when using memory referenced r.m.s. value
	66	ATM command off end of jump table	Command	RAM/ROM failure/corruption.
	67	SSM maths error	Mod No	RAM/ROM failure/corruption.
	68	PCM maths error		RAM/ROM failure/corruption.
	69	ACM range DAC error	Mod No 1	ACM range offset DAC failure, or gain stage failure, or integrator amp. failure, or int. comparator failure, or 5V power failure.
	6A	ACM command off end of jump table	Command	RAM/ROM failure/corruption.
	6B	RTM command off end of jump table	Command	RAM/ROM failure/corruption.
	6C	ACM maths error	MP error no	RAM/ROM failure/corruption.
	6D	ACM illogical threshold and range value		RAM/ROM failure/corruption.
	6E	PCM command off end of jump table	Command	RAM/ROM failure/corruption.
	6F	No power supply fitted	Mod No	Power supply not connected to PCM.
	70	RRM command off end of jump table	Command	RAM/ROM failure/corruption.
	71	SOM command off end of jump table	Command	" " " "
	72	VIM command off end of jump table	Command	" " " "
	73	VIM maths error	Module	" " " "
	74	PGM command off end of jump table	Command	Page 9 RAM/PROM interpreter.
	75	PGM maths error	MP error	RAM or Page 3 error.
	76	TIM Mark/Space neg. width requested		Prog. error due to memory reference or RAM corruption.
	77	TIM NEG/Ovrerrange Hysteresis requested	Mod No	
	78	TIM Ovrerrange threshold requested	Mod No	
	79	TIM Illegal pin no	Mod No	Program storage error or Page 9 RAM error.

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
TEST MODULE HANDLERS	7A	TIM command off end of jump table	Command	Page 9 RAM/PROM interpreter.
	7B	TIM maths error	MP error code	Maths error.
	7C	PCM/PSM, I lim requested for PCM	MOD No	I lim not valid for PCM.
	7D	PSM VOLTAGE/CURRENT out of range	Mod No	Prog. error due to memory reference or RAM corruption.
	7E	PSM COMMAND INVALID	Mod No	RAM/PROM disc error.
	7F			
CONTROL PACKAGE AND GENERAL ROUTINES	80	Illegal test rack module (continuity)	Mod type	Unallowed module in test rack when running an SOM continuity program. Next byte is the module type.
	81			
	82	Buried handler not found	Type byte of switched-in handler	RAM/ROM error.
	83	Call to deleted Grouts subroutine		" " "
	84	Handler not found in soft or hard handler tables		Usually due to module not correctly inserted in test rack. If next byte 80 or greater suspect BSM/PROM failure (page 3), if <80 check test module connectors.
	85	Hard or Soft Handler table full		PROM - Nucleus/Control pages 1 & 2.
	86	PLM failure	Handler status byte*	If status 01 could not find PLM. If status 04 VCC failure. May be 30 volts failed/reversed PROM/short on TPM/PEM hardware failure.
	87			
	88			
	89			
	8A			
	8B			
	8C			
	8D			
	8E			
	8F			
CALIBRATION PACKAGE	90	ATM Trim DAC exceeds 200mV	Mod No	ATM amplifier error.
	91	Module data area not found		Calibration not correctly performed RAM/ROM.
	92	Invalid calibration command		RAM/ROM error.
	93	Previously found PCM now not found		" " "
	94	SSM freq will not trim		SSM freq DAC error or SSM oscillator error.
	95	PCM set to val <0 or >32		Invalid voltage for PCM when using memory referenced voltage.
	96	Maths Pack error		RAM/ROM error
	97	Invalid pin bank (ATM gain)		" " "
	98	PDM gain factors out of range		PDM calibration card not working or ACM input stage failed.
	99	Invalid pin bank	Bank Letter	RAM/ROM error.

* 01 = invalid PLM

04 = Read Mode On failure (30V power supply or 5V switch on PEM fail)

SOURCE	NO	ERROR	ACCUMULATOR CONTENTS	POSSIBLE FAULT
CALIBRATION PACKAGE	9A	PCM iteration count exceeded		PCM unable to attain required voltage. Hardware.
	9B	SSM set to <0V or >8V		SSM asked to produce invalid level when using memory references value.
	9C	Overrange Error in CALPAC	CAL or PCM TRIM	If in CAL use debugger to interrogate stored cal. values.
	9D			If PCM, suspect DAC failure.
	9E			
	9F			
IN-CIRCUIT INTERPRETER	A0	ITM or ZTM Module not found	Module type code	RAM corruption (Pages A-D) / ZTM connectors.
	A1	Maths error	MP error code	Maths error.
	A2			RAM corruption or program medium.
	A3	Overrange on positive drive	Range No	ACM DACOUT failure / ITM hardware / SSM failure / ~~ZTM FAILURE~~ .
	A4	Overrange on negative drive	Range No	ACM DACOUT failure / ITM hardware.
	A5			
	A6	No pulses from SSM	Exception byte	SSM failure/TRC failure (check CTA).
	A7	MAX V >5V or <0.04V		RAM/Program storage corruption.
	A8			
	A9			RAM/PROM
	AA	Overrange during NLCC* measurements (Drive)	Range No	ACM DACOUT
	AB	Overrange during NLCC* measurements (Sense)	Range No	Device under test super diodic.
	AC	MODULE NOT ZTM in GRD or TRI statement	Module type code	Possibly device oscillating.
	AD	Negative Z or R requested in limits		
	AE			
	AF			
IEEE HANDLER (NUCLEUS)	B0	Talker/Listener address number error	Length byte	Program specified too many devices / RAM error.
	B1	Talker/Listener primary/secondary address error	Address	Program RAM / storage corruption.
	B2	Command (IEEE CMD) error	Command No	Program RAM.
	B3	Not sending ATN/REN	Bus status byte	IEEE card failure.
	B4			
	**B5	RMM not fitted		
	B6			
	B7			
	B8			
	B9			
	BA			
	BB			
	BC			
	BD			
	BE			
	BF			

* NLCC = non-linear current component (during semiconductor testing)

** Interpreter

SOURCE

NO ERROR

ACCUMULATOR
CONTENTSPOSSIBLE
FAULT

CONTINUITY INTERPRETER

C CONTINUITY (ZTM) ERRORS

C0	Test/learn flag not recognised at interpreter entry	Flag found (learn = 80) (Test = 8A)	RAM corruption/storage corruption.
C1	Too many ZTMs found in rack while building tables		Test bus (address) failure/ZTM decode failure.
C2	Module not found, or illegal	Mod no. @ Q no. (Hex)	RAM corruption/rack contents changed since program learnt.
C3	Illogical PIN NO in S/C list	No. found	RAM corruption.
C4	R structure corrupt		RAM corruption.
C5	Type code not found (ZTM)	Type code @ H/W address	ZTM failure during program run / bus failure / RAM corruption.
C6	Identical pin Nos in S/C list	Type code @ Q no. (Hex)	RAM/storage corruption.
C7	Maths error		RAM corruption / math chip error.
C8	R val out of range	*	RAM corruption.
C9	Overrange measuring DACOUT	**	DACOUT failure / ZTM failure (driving HI LO).
CA	Error calibrating RODAC	***	ACM failure, range offset DAC / possibly PDM / HI LO jammed.
CB	C structure corrupt		RAM corruption / storage corruption.
CC	C value out of range		RAM corruption / storage corruption / memory referenced value too large.
CD	RODAC threshold out of range 10-120		Corrupted threshold range incorrect; DACOUT error / RAM corruption / maths error.
CE	Cont. Prog. not found		RAM corrupted / storage error.
CF	Cont. Prog. structure failure		RAM corruption / storage error.

D DISC SYSTEM ERRORS

D1	Reset Error	A = Status reg	Floppy disc controller chip error.
D2	FDC Busy	A = FDC command	Floppy disc controller chip error.
D3	Directory Failure: no space for file name		Logical failure on disc / probably RAM error.
D4	Write Non-DMA Error	A = Result reg	Floppy write error. (Disc or Drive) Error type is next byte.
D5	No drive specified in call parameter (software error)	A = 1, Call Parameter	RAM/ROM failure.
D6	Track not found in map	A = Track No	RAM/ROM failure.
D7	Directory Failure: track map error during release		RAM/ROM failure.
D8	Format Non-DMA error	A = Result reg	Floppy format error. Error type is next byte.
D9	Read Non-DMA error	A = Result reg	Read error. Error type is next byte.

* ACC = 0 @ DACOUT No
ACC = 01 NEG. No

*** ACC = 0
ACC ≠ 0

, CODAC SET TO LIMIT OF RANGE
CALIBRATION VALUE NEGATIVE OR
OUT OF ALLOWED RANGE.

** ACC = 0 , TIME
ACC = 1 , VOLTS

SOURCE			ACCUMULATOR CONTENTS	POSSIBLE FAULT
	NO	ERROR		
	DA	Read Non-DMA error when verifying	A = Result reg	Read error. Error type is next byte.
	DB	Command stays busy when non- DMA error	A = Command	Chip error.
	DC	Type Code '00' when Non-DMA error after 10 re-tries	A = Command	Chip error.
	DD			
	DE			
	DF			
	E0			
	E1			
	E2			
	E3			
	E4			
	E5			
	E6			
	E7			
	E8			
	E9			
	EA			
	EB			
	EC			
	ED			
	EE			
	EF			
	F0			
	F1			
	F2			
	F3			
	F4			
	F5			
	F6			
	F7			
	F8			
	F9			
	FA			
	FB			
	FC			
	FD			
	FE			
	FF			

11.0 USER INFORMATION

11.1 Routine checks

The Test Console should be kept clean. No foreign objects, particularly metal ones, should ever be placed inside the Console. During longer periods of non-use, the Console, Control Box, and in particular all Test Jigs carrying spring-loaded pins, should be adequately protected from dirt and damage.

Always check that inter-unit connectors are secure, and ensure that power leads and bus cables do not constitute a safety hazard. Dependent on the amount of use, a weekly cleaning of the print head and guide bar should be performed, using a soft brush and dry tissue. Do not use oil or any other lubricant. Also see para 4.

Six-monthly checks

- 1 Disconnect the equipment from the main a.c. supply. Measure the insulation resistance of each supply terminal to earth. To be not less than $1M\Omega$.
- 2 With the equipment still disconnected, measure the earth bonding continuity from the main earth terminal to each of the points listed below. In every case, the resistance to be not more than 0.01Ω .
 - a) Control Box main frame and cabinet assembly
 - b) Test Console main frame and console casing
 - c) Console assembly mounting bracket
 - d) Printer chassis
 - e) Floppy Disc mounting bracket
 - f) Test Rack (of Console) chassis
- 3 With equipment disconnected from the main a.c. supply:
 - (i) carry out a general inspection of all component parts, including test jigs, for signs of overheating, chafing or excessive wear
 - (ii) check the security of all screw terminals, sockets, pcb's and plug-in modules
 - (iii) dust out inner compartments using a soft brush and vacuum cleaner
 - (iv) remove each pcb and plug-in module in turn and inspect for any sign of general deterioration
- 4 Inspect, and if necessary clean, the cooling fans on the Test Console and on the Control Box. With the equipment switched on, check that both fans are functioning correctly.

11.2 Floppy Disc

Possibly due to manufacturers' advertising, the general impression gained by users is that floppy discs will provide a permanent record of their program.

Floppy disc is an excellent alternative to EPROMS where a large number of programs are to be stored despite the relatively short life and need for careful handling. Wayne Kerr recommend the use of Scotch 744/0 RH 40 Track Diskettes with their ATE systems, incorporating a reinforced hole.

High operating speeds, and surface abrasion from contact with the head, limit the working life of a disc. Experience shows that many discs never attain the normal operating life expectations due to mishandling and bad storage. This is minimised on the Wayne Kerr System, since the disc is rotated only during access, and therefore the vast majority of the time the disc is not under wear.

The following do's and don'ts should be adhered to rigidly to obtain maximum life from your discs.

DO's and DON'Ts for Floppy Discs

- DO ----- Store discs in a dry, dustproof box, preferably anti-static, away from the Workshop environment when not in use.
ONE speck of dust will cause havoc, shortening the disc life considerably.
- DO ----- Return discs to their storage envelopes immediately.
- DO ----- Keep discs and storage box away from magnetic fields: remember heavy duty mains cables can set up a considerable magnetic field.
- DO ----- Insert discs into the drive unit carefully without bending: inserted properly the disc will slide home without undue pressure.
- DO ----- Always keep a master Disc from which further copies can be made.
- DO ----- Use and store diskettes within the temperature range of 50-100°F (10-52°C).

NOW FOR THE DO NOTS

DO NOT ----- Fold or bend the disc.

DO NOT ----- Write on a label after it is stuck on the disc, the pressure may deform the surface.

DO NOT ----- EVER touch the disc face.

DO NOT ----- Leave a disc unprotected on the bench when not in use.
Always return a disc to the protective envelope and storage box when not in use. ----- DIRT AND DUST KILLS!

DO NOT ----- Spill beverages on the disc.

DO NOT ----- FORCE the disc drive door: it will close easily if the disc has been inserted properly. This is the largest cause of drive hole damage and frequently the reason for the system reporting SECTOR NOT FOUND when looking for a program.

When the drive hole is mis-shaped the disc runs eccentrically; consequently the head is unable to find the required program.

DO NOT ----- EVER switch machine on or off with disc inserted in the disc drive. This could damage the disc.

Note: When new, some diskettes are 'sticky', and need to be rotated during loading. If the system is reporting a disc error on a new diskette, insert the diskette into the drive unit, but leave the door open. Invoke any disc operation on the system and, within half a second, close the disc drive door.

11.3 A8315 Debugger

The Debugger is fitted as a standard part of the A8315 firmware set, and complete instructions for its use form a part of maintenance training which is available from Wayne Kerr. Listed here is a summary of the Debugger facilities.

Entry to the debugger is achieved by setting switch S900, inside console, as below:



On pressing LOAD/CAL and RESET simultaneously, the screen is cleared and the following message is printed.

```
A8315 Debugger
>
```

The chevron is the input prompt character. In the text which follows, underlining is shown to identify user typing (and so distinguish these characters, or spaces, from machine output characters). RETURN is used to signify the carriage return/line feed (some keyboards may have alternative labelling for this key). The comments that may now be used are:

D to display memory

```
e.g.      >D 0000 RETURN
           0000 F3 RETURN           RETURN advances forward through memory
           0001 AF ^                 ^ moves backward through memory
           0000 F3
```

If RAM (or Hardware) is displayed in this way, its contents may be changed.
Viz

```
>D A000 RETURN
A000 FF AC  RETURN           RETURN writes to that location and
                                displays next
A001 FE C3  ^                 ^ writes and displays that location
A001 C3 ^
```

Blocks of memory may be displayed by typing in start and finish orders separated by a delimiter, a space or comma. To interrupt listing ESC.
Viz

```
>D 0000 _ 000F
0000 F3 AF C3 80 00 FF FF AF F3 C3 80 00 FF FF AF F3
>
```

Mathematics may also be implemented i.e.

>D 0000 + 001E RETURN

001E AF

N.B. It is not necessary to type all digits in, as leading zeroes are assumed.
If a mistake is made, only the last four digits are accepted.

e.g. >D F RETURN

000F F3

or >D FF33FF4213 RETURN

4213 FF

* In all modes of operation a "." terminates entry and returns the chevron.

P to display port contents. Only one port may be displayed at a time;
the contents may just be monitored or changed as in the D command.
Examples are given below.

>P 80 RETURN

80 60 RETURN

81 03 ^

80 60 61 RETURN

81 03 05 ^

81 05 ^

80 61

C copy memory contents. The Start and End address of the source must be
supplied, also the Start address of the destination. Each parameter must
be separated by a delimiter.

e.g. >C XXXX _ YYYY _ ZZZZ

copies block of data residing at XXXX to YYYY, to block of memory ZZZZ
to ZZZZ + (YYYY-XXXX).

B puts a software breakpoint at the address specified. When the break is
encountered, the Program Counter and AF registers are O/P and the chevron
is returned.

If BXXXX,1 is I/P, all registers are displayed at break.

- E executes the program from the specified location. If no parameter is I/P it executes from present user Program counter.
- R displays all user registers. If no parameter is I/P, only the registers are displayed. If 1 is entered as a parameter, headings are also O/P.

* N.B. Registers are only valid when a breakpoint is encountered, i.e. system registers are copied to RAM on breakpoint entry. The register may be changed.

When executing an E command, the saved registers from RAM are copied back into the CPU registers, and then the user program has control.

With all Memory referenced commands, Mnemonics may be used instead of memory locations. These must be preceded by /.

Resident Mnemonics are:

/PC	User Program Counter	/A'	User alternative A reg		
/SP	User Stack Pointer	/F'	"	"	F "
/A	User A reg	/B'	"	"	B "
/F	User F reg	/C'	"	"	C "
/AF	User AF Pair	/D'	"	"	D "
/B	User B reg	/E'	"	"	E "
/C	User C reg	/H'	"	"	H "
/BC	User BC Pair	/L'	"	"	L "
/D	User D reg				
/E	User E reg				
/DE	User DE Pair				
/H	User H reg				
/L	User L reg				
/HL	User HL Pair				
/I	User Interrupt Vector Register				
/IF	User Interrupt Flip Flop 00 = Unset 04 = Set				
/IX	User IX reg				
/IY	User IY reg				
/CO	Console - out driver address				
/CI	Console - in driver address				
/RT	Ramtest Program Address				
/VT	Vim Test Program				
/DT	Disc Test Program				
/ST	Software Test Program				
/SY	Re-enter System				

A8315 Functions Available

1 Switch in Cal Package by storing Hex 12 at 3000.
This can be checked by monitoring 3000 & 3001 which should be 'FC' & 'FC'.
Executing 3005 causes calibration values of ACM to be output to screen.

2 Executing /SY takes one back to READY.

3 Executing 0 re-enters A8315 system from start.

4 Software Tester

Executing /ST automatically enters the software tester, which first of all tests the system RAM. If the RAM passes it displays

RAM OK

The software tester then asks which software requires testing, to which the reply is Y (Yes) or N (No). The options are BASIC SYSTEM, PROGRAMMING OPTION, ADD OPTION. If the software is correct, then OK is O/P against the relevant heading, if any failures are detected, the number of the failing PROM, its board location and the NEW signature is displayed.

Note: If the Debugger has failed, the software tester EPROMS can be installed in the sockets BSM5 1215 and 1216 and the test initiated by depressing LOAD/CAL and RESET. (In this case three failures will be O/P corresponding to BSM5 sockets 1213, 1215, 1216).

N.B. Debugger resides at Page 4 BS '1E'

Debugger Ram is at top of Page D.

Therefore Debugger can test everything except Block switching in Page 4.
Ram Test can be used to test Debugger Ram as the test is non-destructive.
Use of the Debugger will corrupt any learnt ZTM Continuity program held in RAM.

Continual use of the Debugger may, in some circumstances, give rise to SYSERR 08 due to system stack overflow. Reset will clear this condition and allow continued use of Debugger.

11.4 Test Jig Accessories

The pins and tools recommended for bed-of-nails operation are listed below.
The Forwessun 'Probe Tech' types come in two versions :

- a) Short stroke for mechanical hold down
- b) Long stroke for vacuum hold down.

Mechanical hold down parts		Vacuum hold down parts	
Short Stroke Receptacle	PT 170/2WS	Long Stroke Receptacle	PT 200/2WS
Serrated Probe	PT 707SE/1	Serrated Probe	PT 101SE/1
Needle Probe	PT 703SP/1	Needle Probe	PT 103SP/1
Cup Probe	PT 706CC/1	Cup Probe	PT 104CC/1
Receptacle Installation Tool F 170		Receptacle Installation Tool F 200	

Note that the installation tool is essential, as fitting of the receptacles involves deformation of a compression ring to give a more positive location.

Whatever pins are used, care must be taken to ensure that their use with the mechanical hold-down allows the locking forks on the frame to engage prior to the frame being stopped by resting on the board under test.

Repeated abuse of the hold-down system may result in invalidation of the system warranty.

11.5 VDUs in use with the system

Several types of VDU are in current use with the system, all using standard 64-character ASCII keyboards, which are compatible. However, the machine control codes are not compatible.

If a VDU requires servicing, every attempt will be made to replace it with one of the same make, but this may not always be possible.

If an alternative make of VDU is supplied, Wayne Kerr cannot be held responsible for any inconsistencies in programming or machine lock-ups during a program run, neither will they be held responsible for loss of production, production time or any expenditure incurred in the reprogramming that may be required.

11.6 Transit Packaging

UK Transit

- 1 CONSOLE packaging consists of a three-ply wood transit case fitted with two carrying handles. The case is lined with:

medium-density plastic foam having an average thickness of 75 mm to accommodate and protect the console. Shaped plastic foam covers the console and is held in firm contact by the case lid which is secured in place by 12 x 25 mm screws.

Case dimensions 80 x 75 x 38 cm.

Weight 42 kg gross.

- 2 CONTROL UNIT is packed in a two-piece close moulded medium-density plastic foam liner, inside a triple walled cardboard box, closed at the small end with plastic tape.

Box dimensions 75 x 75 x 37 cm.

Weight 56 kg gross.

- 3 VDU packaging is of the double container type. The inner cardboard container has a moulded medium-density plastic foam liner to prevent movement of the VDU, and this container is in turn inserted into the outer triple wall transit box.

The inner container is supported at each corner by firm plastic foam angle pieces, providing a 75 mm space between inner and outer cases in each direction.

Case dimensions 70 x 55 x 50 cm.

Weight 20 kg gross.

The System is transported to customers in the UK by Wayne Kerr or by carrier as three separate cases as described.

Overseas Transit

Details vary according to shipment and destination.

12.0 SUPPORT & SERVICES

Wayne Kerr provides a customer service covering

TRAINING courses to introduce users to programming and testing with the ATE,

SUPPORT for technical applications and advice on ways to meet individual test requirements,

SERVICE in the unlikely event of breakdown, to get a system rapidly back in operation.

MAINTENANCE agreements for a system and documentation update facility, together with a priority servicing arrangement.

(TECHNICAL: NIK CLARE EXTN: 21581)

ENGLAND:

(Training Instructor - Dave Chapman)

W K R LIMITED
DURBAN ROAD,
BOGNOR REGIS,
WEST SUSSEX,
PO22 9RL.

TELEPHONE NO: BOGNOR REGIS (0243) 825811
AFTER OFFICE HOURS: (0243) 827195
TELEX NO: 86120 WAYKER G

GERMANY:

WAYNE KERR GmbH,
DOMSTRASSE 58-62,
D6050 OFFENBACH/MAIN,
FRANKFURT,
WEST GERMANY.

TELEPHONE NO: 0611 - 810778
TELEX NO: 4185470 WAYKR D

USA:

WAYNE KERR INC.,
400 WEST CUMMINGS PARK,
WOBURN,
MASSACHUSETTS 01801,
USA.

TELEPHONE NO: (617) 938 - 8092
TELEX NO: 6817257 A/B: WKERR WOBN

A.0 APPENDIX A: TEST MODULE SPECIFICATIONS

A.1 Analog Test Module (ATM)

FUNCTION	Basic analog measurement facilities (including verify and adjust operations) of voltage, current, frequency, events, period and pulse width.		
PROTECTION	All ATM pins: up to 100V (peak, ref. analog ground)		
INPUT RESISTANCE	1MΩ ± 0.3%		
DC VOLTAGE (+ve or -ve)	Ranges (FS):	1 0.5V	5 8V
		2 1V	6 16V
		3 2V	7 32V
		4 4V	8 64V
	Range 8 can be extended, using an external resistor		
	Resolution:	Range FS x 25 x 10 ⁻⁶	
	Accuracy:	0.1% of reading, or 0.05% of Range (whichever is greater)	
	Integration time:	Programmable 100µs to 8s (resolution 0.5µs) Default value 20ms	
	CMRR:	60dB (using A & C pin pair)	
AC VOLTAGE	Ranges & Resolution as for DC		
	Accuracy (Extended Bandwidth, EB)		
	20Hz - 20kHz	±0.1dB of reading or 2.5mV (whichever is greater)	
	1Hz - 20Hz	} ±1dB of reading or 2.5mV (whichever is greater)	
	20kHz - 200kHz		
	Accuracy (Normal, low-noise, bandwidth)		
	-1dB at 20Hz and 20kHz		
	-3dB at 40kHz		
	-14dB at 200kHz		
	CMRR:	60dB to 1kHz, derate by 20dB per decade frequency	
	Note:	200Hz ACM option reduces settling time. Accuracy is then 0.1dB at 200Hz; 1dB at 25Hz. (HF as before).	
PEAK VOLTAGE/ TROUGH VOLTAGE	(Verification that input signal exceeds a specified threshold within a specified gating time).		
	Gate time maximum: 8s (Min I/P 1 cycle, p-p ≥ 20% f.s.)		
	Voltage Accuracy:	0.5% of full scale ± 5mV	

DC CURRENT Computed by system from voltage measured across external resistor (value optimised for permissible voltage drop in circuit of UUT). Accuracy determined primarily by resistor tolerance.

FREQUENCY Multiple periods are measured automatically, optimising accuracy and speed. Program can specify 16 or 256 periods. By default, system will use 1 period for frequencies below 200Hz, 16 unidirectional transitions above 200Hz, 256 transitions above 13.2kHz, and a gate time of 32ms above 126.6kHz.

Range: 20Hz - 200kHz

Accuracy: Better than 0.05%

Threshold: AC coupled or Programmed DC level
(0 to $\pm 64V$)

PERIOD	An alternative to frequency for low frequencies or for cyclic signals
--------	---

Maximum: 8s

Resolution: $0.5\mu s$

Accuracy: $0.01\% \pm 0.5\mu s$

Transition: positive or negative

Threshold: AC coupled or Programmed DC level
(0 to $\pm 64V$)

PULSE WIDTH	Maximum:	8s
	Resolution:	0.5μs
	Accuracy:	0.01% ± 0.5μs
	Polarity:	positive or negative
	Threshold:	AC coupled or Programmed DC level (0 to ± 64V)

EVENTS	The occurrence of specified transitions can be recorded or verified within a specified gate time
--------	--

Gate time maximum: 8s

Gate Accuracy: $0.01\% \pm 0.5\mu s$

Count Accuracy: +0, -1

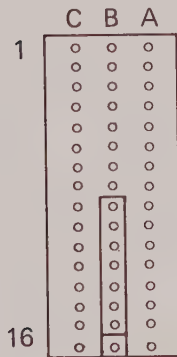
Transition: positive or negative

Threshold: AC coupled or Programmed DC level
(0 to $\pm 64V$)

Count Maximum: 65,535

INPUTS* Single-ended: 30. Reference is system ground level
 Differential: 15x15 (i.e. 225 max)

CONNECTIONS Inputs*: pins A1-A15 & C1-C15
 Address: " B9-B15 (B16 = address ground)
 Pseudo Signal Ground (PSG) pin A16
 Analog Signal Ground (ASG) pin C16



* Single-ended inputs are pins A1 - A15 and C1 - C15.

Differential inputs are any A pin reference to any C pin, reducing the number of pins available for single-ended inputs. In programming differential inputs, pin A should be specified before associated pin C.

Measurements can be made on specified ranges or, if no range is selected, the system will autorange.

A.2 POWER SUPPLY MODULE (PSM)

FUNCTION	Control of independent, fully floating, programmable power supplies. Three PSMs can be fitted, making six supplies available. The PSM can replace the earlier PCM with no changes to wiring or program, but the PCM cannot replace a PSM directly.
VOLTAGE	0-32V. Resolution: 10mV. Accuracy: 0.1%±5mV. Outputs can be strapped positive or negative to ground or stacked in series for voltages exceeding 32V. Program reference is to module 1/A, 1/C, 2/A, 2/C, 3/A or 3/C.
CURRENT	Limit (1.1A max.) set by ILIM field of program (see 9.9.10) or by resistor. See footnote* and graph overleaf. Accuracy of programmed limit is 6%±20mA, resolution 5mA. If limits are set by program and by resistor, the lower one will apply. If no limits are specified, default value is 1.1A. Power supplies may be connected in parallel to increase current capability, but whenever this is done it is essential to limit current on all supplies to 1.0A (see 9.9.11).

CONNECTIONS



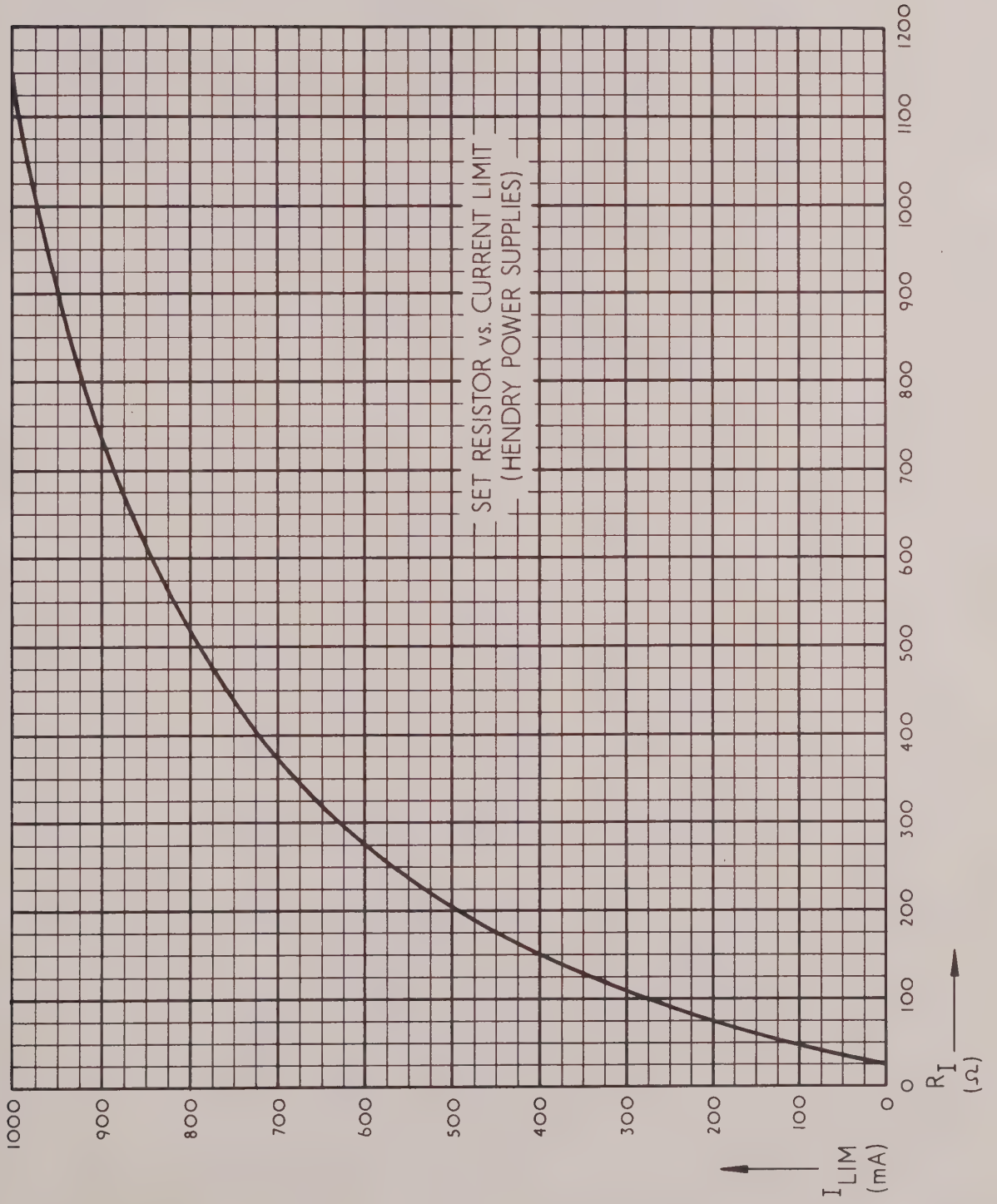
- A Supply -ve : pins A1-A6 (commoned)
- " " +ve : " A9-A14 "
- " " current set resistor between A15 and A16
- C Supply -ve : pins C1-C6 (commoned)
- " " +ve " C9-C14 "
- " " current set resistor between C15 and C16
- Calibration reference voltage : pins B1/B2 (commoned)
- Restricted address (1,2 or 3) : pins B14 and B15
- Address Ground : pin B16

When the program removes a supply (REM PSU), outputs are relay isolated, allowing calibration of ATM pins directly connected.

* On Hendry power supplies, for current limit up to 1A, use set resistor of value given by

$$R_I = \frac{249I_{LIM} + 28.67}{1.2417 - I_{LIM}}$$

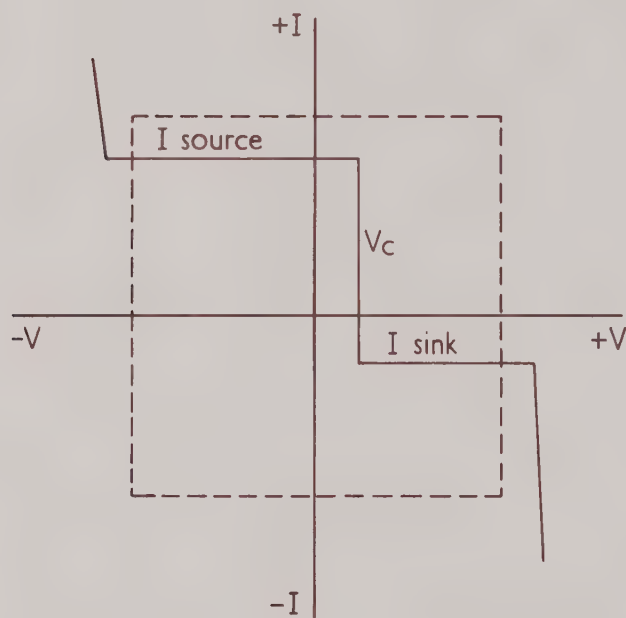
where R is in ohms and I is in amps.



A.3 Voltage/Current Source Module (VIM)

FUNCTION Dual programmable DC voltage/current force stimulus. Operation is in constant-voltage or constant-current mode according to the programming of the three parameters: voltage; current source; current sink.

OPERATION The permissible area of operation is outlined by dotted lines on the graph. The continuous line is a typical programmed function.



PROTECTION Catching fuses limit I max to 0.5A @ 15V nom.

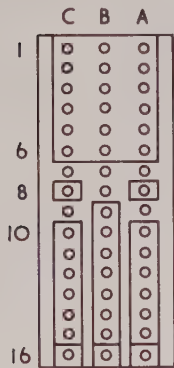
VOLTAGE

Range:	0-12.8V +ve or -ve
Resolution:	3.125mV
Accuracy:	±0.5% ± 5mV
Output resistance:	<0.5Ω (remote sensed) <1.5Ω

CURRENT

Ranges:	0-128mA (source+, sink-)
Resolution:	30μA
Accuracy:	±1.5% of setting ± 50μA
Output conductance:	<10μS
Monitor output:	Scaled at -40mV per mA Accuracy: ±0.5% ± 1mV

CONNECTIONS



A Supply output:	pins A10-A15
" " current monitor:	pin A8
" " remote sense:	pin A16
C Supply output:	pins C10-C15
" " current monitor:	pin C8
" " Remote sense:	pin C16
Ground (APG):	pins A1-A6, B1-B6, C1-C6
Address:	pins B9-B15 (B16= address ground)

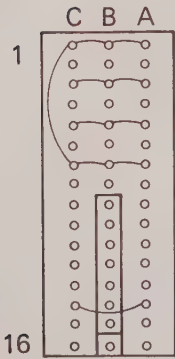
Note: Current monitor outputs can be connected to an ATM pin, but the pin must be isolated for calibration (see WARNING in section 4.0).

A.4 Sinewave Source Module (SSM)

FUNCTION	Analog stimulus, programmable in frequency and output level, with d.c.-coupled and a.c.-coupled sinusoidal outputs and an unswitched square-wave output.
FREQUENCY	Range: 20Hz - 200kHz
DISTORTION	20Hz - 50kHz: <0.25% 50kHz - 200kHz: <2.5%
LEVEL	0 - 8V rms in 1mV increments Accuracy (1kHz): 0.1dB ± 16mV above 100mV. 0.3dB (typically) below 100mV, 1.5dB max
FREQUENCY RESPONSE	0.05dB (20Hz - 200kHz)
OUTPUT IMPEDANCE	1Ω + 20μH
CURRENT	Maximum 10mA
AC OUTPUT	Coupled via 470nF
SQUARE-WAVE	5V peak, unswitched output, synchronised with the sinusoidal output
PROTECTION	Fully protected against short-circuits to ground or to the internal power supplies. On-board fuse protection for higher overloads

Note: Unused SSM outputs (i.e. quiescent) present an impedance of 16Ω and, being relay isolated, will tolerate 64 volts.

CONNECTIONS	DC-coupled output: pins A3, B3, C3 (linked)
	AC-coupled output: pins A5, B5, C5 (linked)
	Square-wave output: pins A14 and C14 (linked)
	Analog ground: pins A1, B1, C1, A7, B7, C7 (linked)
	Address: pins B9 - B15 (B16 = address ground)



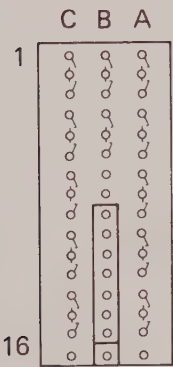
A.5 Reed Relay Module (RRM)

FUNCTION Application and routeing of signals, load switching isolation and by-pass functions. Each RRM consists of 24 reed relays, connected in 12 groups of 2. Since each relay is individually controllable, they can be formatted by software into matrices, i.e. 4x6, 6x4, 2x12 etc. for switching functions. Alternatively they can be driven from tables to simulate front-panel switches.

	<u>Dry</u>	<u>Mercury wetted</u>
MAXIMUM SWITCHING VOLTAGE	64V	64V
MAXIMUM CURRENT (AC or DC) SWITCH	0.5A	1A
MAXIMUM CURRENT (AC or DC) CARRY	0.5A	2A
MAXIMUM POWER	10W	50W
MAXIMUM CONTACT RESISTANCE	0.25Ω	0.25Ω
OPEN RESISTANCE	>1GΩ	>1GΩ

CONNECTIONS Common line to each pair of relays (see diagram) is pins 2, 5, 8, 11 and 14 of banks A and C; pins 2 and 5 only on bank B.

Address: pins B9-B15 (B16 = address ground)



A.6 Timer Interface Module (TIM)

FUNCTION Timing measurements on analog signals on one channel or between two channels with full choice of start and stop polarity combinations; event counting and frequency measurement on either channel; measurements of delay, phase relationship and voltage slope. Threshold and noise immunity are programmable.

MINIMUM SIGNAL STEP 150mV

MAXIMUM INPUT 100V

INPUT RESISTANCE 200k Ω $\pm$ 0.1%

THRESHOLD Setting: -32V to +32V
Resolution: 16mV
Accuracy: $\pm$ 1% $\pm$ 50mV

NOISE IMMUNITY Setting: 120mV - 1.6V ($\pm$ 7%)

BANDWIDTH 1MHz

TIMER Range: >35minutes
Resolution: 0.5 μ s
Accuracy: 0.01% $\pm$ 0.5 μ s

CONNECTIONS



Inputs: pins A1,A4,A7,A10,C1,C4,C7,C10
Ground (APG): pins A2,A3,A5,A6,A8,A9,
A11-A16
" " " C2,C3,C5,C6,C8,C9,
C11-C16
" " " B1-B8
Address: pins B9-B15 (B16= address ground)

(Performance is quoted for noise-free inputs)

#420744

A.7 Tristate Impedance Test Module (ZTM)

FUNCTION Verification of impedance values, within a circuit, with 32 pins which may be individually or collectively designated as guards or tristated. When all pins are tristated, the UUT may be functionally tested provided no applied voltage exceeds $\pm 15V$. Additionally, when used in the A8315 Combat system, the ZTM can perform high-speed continuity testing using all 32 pins. Continuity programs are learned from a known good board and each pin is tested against all others and to system ground. For any ac impedance tests, an SSM must also be used and must carry address 4. Any LTM or DTM pins remaining connected to ZTM or ITM pins will make in-circuit measurements inaccurate. Consequently, digital test pins should be disconnected, using RRM, when in-circuit testing is being done.

Unguarded Impedance Measurements

RESISTANCE	Range:	$3\Omega - 10M\Omega$ (measured at DC)
	Accuracy:	$\pm[(1 + 2Rx10^{-6})\% + 0.12\Omega]$
CAPACITANCE	Ranges:	$50pF - 1000pF$ (Z measured at 1kHz) $1000pF - 8\mu F$ (Z measured at 150Hz)
	Accuracy:	$2\% + 20pF$
ELECTROLYTICS	Ranges:	$0.1\mu F - 160\mu F$ (Z measured at 100Hz) $160\mu F - 800\mu F$ (Z measured at 20Hz)
	Accuracy:	2%
INDUCTANCE	Range:	$8mH - 8kH$ (Z measured at 200Hz)
	Accuracy:	$(2 + L/400)\%$
		Note: DC resistance must exceed 1.5Ω
IMPEDANCE	Range:	$10\Omega - 1M\Omega$
	Accuracy:	$2\% + (0.75\% \text{ per } 100Hz \text{ above } 300Hz)$
	Test Signal:	Programmable frequency

Semiconductors

Diodes, biodes (reverse-connected parallel diodes) and transistors can be parametrically tested (see sections 9.5.9 to 9.5.13). Use of tristate pins

allows multiple device testing to be performed easily. Parallel resistance of 500Ω is acceptable and 150Ω to a Guard pin from the Drive pin.

Guarding

All pins have a guarding capability with a guard impedance of less than 0.15Ω . On each module, 7 pins may be used in double guard mode, enabling 6-terminal measurements to be made. Guarding impedance is then less than 0.015Ω . Further information on guarding is in sections 9.5.1 and 9.5.8.

Drive/Guard Current

Drive/Guard current must always be limited to 5mA by, if necessary, reducing drive level using false limits. Refer to section 9.5.2.

Tristate

Each pin may be individually specified to be tristate, i.e. 0/C, or guard during in-circuit measurements. While functional testing is being performed, all pins should be specified tristate.

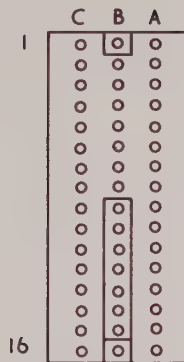
Tristate load	25pF fixed capacitance
Tristate voltage range	$\pm 15\text{V}$ absolute maximum
Current drawn	Maximum 200nA (typically 50nA)

Continuity (Combat A8315 systems only)

All 32 pins are used and the program is self-learning. The continuity checks are embedded in a normal functional and/or in-circuit program. Eight ZTMS will typically pass a board in $2\frac{1}{2}$ seconds.

Threshold resistance	Default: 10Ω
	Specified: 3Ω to $5\text{k}\Omega$
Threshold accuracy	$\pm(10\% + 1\Omega)$
Maximum board capacitance	Default: 0
	Specified: 0 - $1000\mu\text{F}$
Speed (zero capacitance)	10ms/pin (300ms/module)

CONNECTIONS



AC Input (from SSM):

pin B1

Test (Sense or Source):

pins A1-A16, C1-C16

Double pole test:

pins B2-B8 with C2-C8
respectively, referred to
by C numbers

Address:

Pins B9-B15

Address ground:

B16

A.8 Pulse Generator Module (PGM)

FUNCTION	Provision of a defined number of pulses, or a continuous train, programmable for mark and space times and for level. Its output can also serve as a timing signal for control purposes.
PROGRAMMABLE OUTPUT	Level (Mark & Space): -12.8V to +12.8V Resolution: 7mV Accuracy: $\pm 1\% \pm 35\text{mV}$
TTL OUTPUTS	Four supplementary open-collector outputs at TTL level (mark/space times as programmable output) with independent control for enable/disable and for polarity.
RISE/FALL TIMES	<300ns (max output into open circuit) <100ns (at TTL level)
OUTPUT CURRENT	$\pm 10\text{mA}$ up to $\pm 10\text{V}$, falling linearly to $\pm 8\text{mA}$ at $\pm 12.8\text{V}$
NUMBER OF PULSES	Programmable 1 - 65,535 or continuous
MARK/SPACE	Max. ratio 32,768 : 1
TIMING	Ranges, resolution and accuracy: Eight binary ranges, of resolution 0.5 μs , 1 μs , 2 μs , 4 μs , 8 μs , 16 μs , 32 μs and 64 μs . Max. pulse width: resolution x 65,535 Accuracy: $\pm \frac{1}{2}$ resolution $\pm 0.05\%$ Min. pulse width: 2 x resolution Range applies to both time intervals (mark & space) and will depend on the longer interval.
TRIGGER	Immediate (from executable statement), or from System timing signals: CTB from LTM or TIM only. CTA similar, normally high; when low indicates integrator ramp-down during analog measurements. Also used to set-up SSM. GTO normally high; when low, indicates integrator ramp-up or gate time for Events.

CONTROL

The PGM output can be configured to mimic the timing (not the levels) of the CTA, CTB or GTO lines.

CONNECTIONS

Programmable Output: A1/C1 (linked)

APG: A2/C2 "



```
TTL Outputs:  Out      Invert    Enable
```

1: C3 C4 C5

2: A3 A4 A5

3: C6 C7 C8

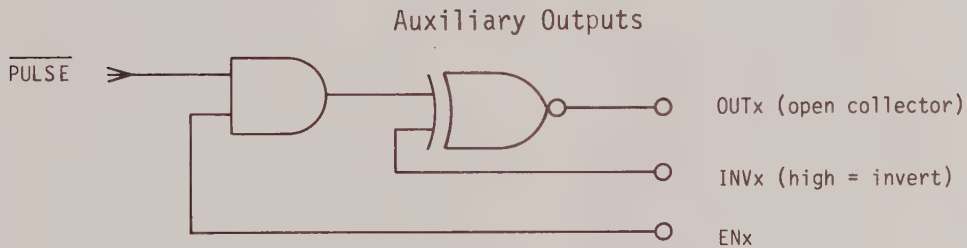
4: A6 A7 A8

APG also on B1 - B4

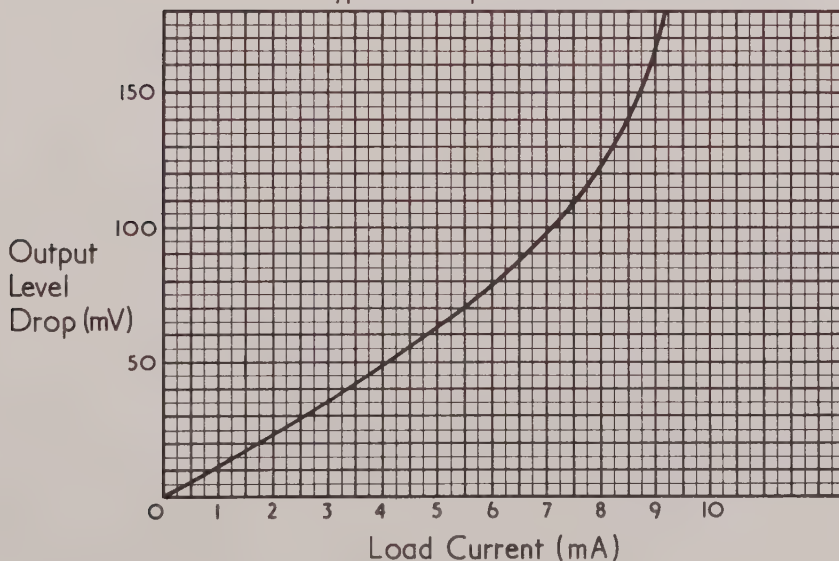
DGND on A9 - A16, B5 - B8 and C9 - C16

Address: B9 - B15

Address Ground: B16

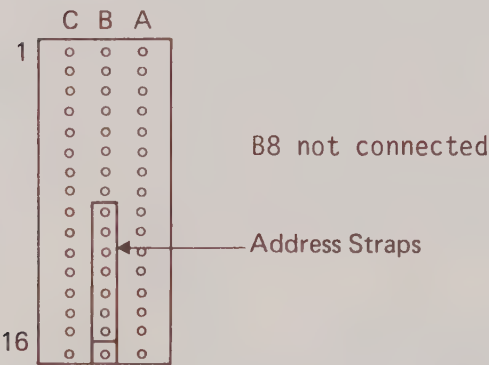


PGM Typical Output Characteristic



A.9 Shorts & Opens Module (SOM)

FUNCTION	To construct a continuity test program by a self-learning process from a known good board. (Continuity Programming is described in section 9.3).
TEST POINTS	39 pin connections per SOM and up to 13 SOMs at one time (i.e. a total of 507 test points).
S/C and O/C	Resistance value separating shorts from opens is programmable from 5Ω to 50Ω (±5%±1Ω). This corresponds to test signal levels from 50mV to 500mV.
CHARGING DELAY	Circuit boards with capacitors are automatically time scaled by the software. Programmer specifies maximum capacitance present (Default assumes 4μF). Values up to 0.25 Farad can be handled.
CONNECTIONS	See TIC diagram (below).



A.10 Logic Test Module (LTM)

≠ 275

FUNCTION	To provide functional testing of digital circuits (and digital tests on hybrid circuits) by sampling actual pin levels against programmed pin levels. Digital programming is described in section 9.7.		
TEST POINTS	Each LTM provides 16 programmable driver/sensor pins. When driving, pins emulate any ground-based logic family up to 15V. This is definable per module by linking the high drive threshold, V_{DH} , (e.g. the logic power supply) from the test jig to a special module input pin.		
LOW DRIVE THRESHOLD	<0.4V at -16mA max		
HIGH DRIVE THRESHOLD	$V_{DH} \pm 0.1V$ ($0 < V_{DH} < 14V$)		
DEFAULT (not linked)	TTL : 3.0±0.1V		
HIGH DRIVE IMPEDANCE	1kΩ		
SENSE THRESHOLD*	0-13.5V, programmable per module Resolution: 53mV Accuracy: ±50mV		
INPUT CURRENT	<50µA		
SWITCHING	Each pin may be switched to CTA or CTB (system counter inputs) to measure time AA,BA,BB or AB. Only one pin may be switched to each at one time. Time (AA,BA,BB or AB) Max in 35 minutes Accuracy ±0.5µsec Events (single input) Max Count 4x10 ⁹ Max I/P Freq 1.5MHz Accuracy +0,-1 count Frequency (single input) Max Freq 1.5MHz Accuracy ±0.025%		

* Sense thresholds and programming is the same as for DTM.

CONNECTIONS

See TIC diagram (below).



The 16 digital pins appear on 1 - 16 on both A and C banks, as shown for the first three. Program reference may be to either bank.

- B1, B2 High Drive Threshold Input
- B3-8 Digital Ground
- B9-15 Address Link pins
- B16 Address Ground

A.11 Digital Test Module (DTM)

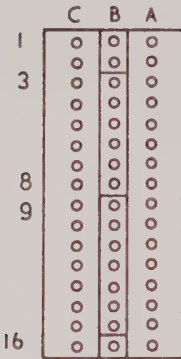
+365.

FUNCTION	To provide functional testing of digital circuits (and digital tests on hybrid circuits) by sampling actual pin levels against programmed pin levels. Digital programming is described in section 9.7.
TEST POINTS	Each DTM provides 32 programmable driver/sensor pins. When driving, pins emulate any ground-based logic family up to 30V. The high drive supply must be provided by connections in the jig to the relevant programmable supply. Each pin has a fixed pull-up resistor of 15k Ω to this supply.
PROGRAMMING	By incremental pin state change. Permissible states are: High/Low/Invert/Clock Drive/Sense Ignore In addition, symbolic definitions of a 'name' against a pin list allow the later equating of that 'name' with a pattern, i.e. truth table patterns. Digital data input from a 'name' to a system store may be performed, allowing PROM checking, etc. Patterns may be incremented or decremented.
TESTING	Sampling may be global (sets Pass/Fail) or local (sets transitory flags). The latter provides conditional testing, e.g. for initialisation and diagnostics.
LOW DRIVE THRESHOLD	<0.4V at -40mA max
HIGH DRIVE (permanent)	15k Ω to test jig supply
SENSE THRESHOLD	Programmable as single level or dual level. In the latter case, the system software automatically performs dual level samples. 0 to 13.5V, programmable per module Resolution: 53mV Accuracy: $\pm 100\text{mV}$

INPUT VOLTAGE -0.5V to +30V
LIMITS

(No timer switching is available)

CONNECTIONS The 32 digital test pins are all of the A and
 C pins



- B1, B2 High Drive Supply
- B3 - B8 Digital Ground (must be connected to UUT)
- B9 - B15 Address Link Pins
- B16 Address Link Common

A.12 Clock/Calendar Module (CCM)

This option, installed in the Control Box, provides test program access to time and date information, primarily intended for inclusion on the test result 'ticket' for quality control records.

The test program can make separate access to date and time information and to separate components of the time. Subsequently, this information can be printed, displayed on the VDU, sent out through the auxiliary SIO port, etc.

The module is non-volatile, maintaining itself when the system is not powered, through the use of a primary battery.

Facilities are provided within the programming option for setting the time and date into the module for enhancing the long-term accuracy and initial setting after a battery change.

4 year Calendar with leap year correction.

Time in hours, minutes and seconds.

Accuracy: 10 seconds per month.

Battery life: 5 years

European format:	25/12/83	}	Selected according to 50/60 Hz link.
American format:	12-25-83		

A.13.1 Program Link Module (PLM)

The Program Link Module (PLM) provides the interface between the user's jig or Test Program Module (TPM) and the Test Bus. Its facilities

- allow the system to differentiate between EPROM and DISC resident programs

- allow the system to read a program from Disc (see section 5.0)

- allow the system to read the program from EPROM

- allow the program to be 'blown' onto EPROM in conjunction with a PEM (A8000 only)

- protect users PROMs against system faults (A8000 only).

A.13.2 Prom Enable Module (PEM) [A8000 only]

The Prom Enable Module provides the programming facilities for the PLM.

A.14 High Speed Extended Maths (Option on A8000)

The extended maths option provides the following additional mathematical functions:

↑	power	
EXP	power of "e"	
SIN	}	angle input in degrees
COS		
TAN		
ASIN	sin ⁻¹	}
ACOS	cos ⁻¹	
ATAN	tan ⁻¹	
LOG	base 10 logarithm	
LN	natural base logarithm	

The option performs all mathematical functions at high speed and, because mathematics are used extensively while testing, it increases the system testing rate.

Note that, in some instances, operands must be in parentheses.

A.15 Mini-Disc (Option on A8000)

This consists of a mini-disc drive (5.1/4" floppy), mounted with its interface in the test console, and the software interface package.

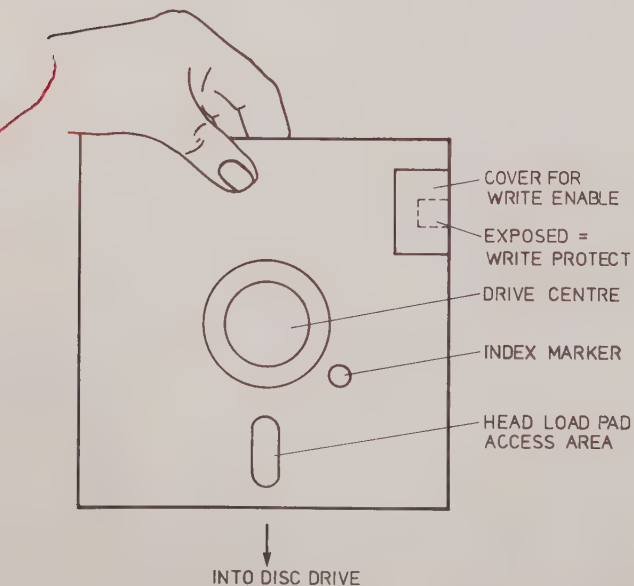
It allows floppy discs to be used as the program storage medium, and can act as an alternative medium to EPROM for A8000. The relevant program on the disc can be identified by a linking plug from the test jig to the program interface. Multi-program jigs are easily implemented by making these links switchable (e.g. by thumbwheel switches etc).

A8000 and A8315 can be supplied as programming and testing stations, or as test only stations. All A8315's, however, have a mini-disc unit fitted (on A8000 the unit is optional). Programs written to EPROM (i.e. on A8000) can be run on A8000 or on A8315. When ordered as programming stations, A8315's (and A8000's with the mini-disc option fitted) allow programs to be written to disc during editing.

Each disc is organised in 37 tracks, each track being equivalent to one EPROM (2K bytes). Maximum program size is 8 tracks. A typical program line occupies 15 bytes.

Copying of discs on a single drive unit may be performed by sequential reading and writing of individual programs. An extension drive can also be provided, permitting fast image copying of the whole disc.

Note A cut-out in the floppy disc sleeve provides write protection, and must be covered to obtain write enable.



IMPORTANT Disc identification labels should be written before attaching them to the sleeve. If labels are already attached, use only a soft (e.g. felt-tipped) pen: NEVER use a ball-point.

A.16 SIO Options

The SIO module can be used as an RS232C port to external devices in two configurations:

PORT 1 (VDU) This option is supplied as part of the Test Station option and is the programmer's terminal. However, it may be installed without the TPG software and supports the IN P1 and OUT P1 statements in the test program for run-time data communication.

PORT 2 (AUX) This option responds to IN P2 and OUT P2 test program statements in a similar manner to Port 1 (for differences, see Appendix B2). It may be operated in full hardware handshake mode.

When the programming option is fitted, it also allows:

- 1 Printing of programs on an external printer
- 2 Transmission of programs to an external device, e.g. A8000 or word processor
- 3 Reception of programs from an external device.

No extra software is required; the user commands are included in the programming option.

Both configurations support V24 (RS232C) interfacing as standard and may be modified to support 20mA current loop. Simple switch settings allow the modules to be used as the computer or peripheral end of the communication link. Data formats supported:

Baud rate:	110,150,300,600,1200,2400,4800,9600.
Length:	7 or 8 bits
Stop bits:	1 or 2
Parity:	None, Even or Odd
Hardware Handshake :	None Port 1. Optional Port 2

A.17 IEEE Control Interface

This option, installed in the Control Box, provides a control capacity for the "IEEE Standard Digital Interface for Programmable Instruments" (IEEE Std 488-1978) [equivalent to IEC-625].* This enables the system to operate and receive data from instruments which also conform to this standard.

The system software gives the user transparent access to the IEEE interface to permit full flexibility of instrument programming formats, as well as high level 'standard' interface commands. Comprehensive string handling facilities are also provided to enable intelligent processing of results, required to cope with the wide variety of data codes and information formats.

The automatic conversion routines provide resolution to $7\frac{1}{2}$ decimal digits.

The controller supports instruments with extended (secondary) addresses.

Up to 14 instruments may be controlled directly from the system. This number may be increased by the use of a proprietary bus extender.

* For those familiar with this standard, the subsets provided by this option are:

SH1,AH1,T4,TE0,L2,LE0,SR0,RL0,PP0,DC0,DT0,C1,C2,C3,C4,C28

A.18 Automatic Digital Diagnostics

Full details of this option are given in the Supplement.

In conjunction with the standard digital testing facilities (LTMs and DTMs), the system can be fitted with this software option which provides intelligent guided probing of digital circuits and automatic fault location.

The internal nodal data is captured and stored as a compressed 'signature'. Sophisticated algorithms ensure that faults within feedback loops will not cause indefinite cyclic probing. No separate input of the test boards 'image' is required, the connectivity information required for later diagnosis is deduced at 'signature' learn time.

After first creating a test program to perform PASS/FAIL testing of the board, utilising its external connections, the system can be put into 'signature' learn mode. The user can then input the internal nodes at which signatures are to be learnt. During this process the system determines the connectivity of the internal nodes (and presents this information for simple confirmation).

If at any time, a node is probed and found not to be 'moving' during the operation of the test program, the user will be warned. This provides a highly effective means of determining the quality and effectiveness of the basic test program.

It is not necessary to learn 'signatures' and connectivity for the whole test board; small areas can be learnt selectively and added to if necessary. This ensures a high degree of cost-effectiveness, when compared to the 'blunt' all-fault diagnostics. 'Signature' verification can also be performed, permitting detection of inconsistent 'signatures', and permitting modification of the test program without the whole learning sequence having to be repeated.

During testing, if a fault is detected from the basic PASS/FAIL test program, the system automatically switches into diagnostic mode, and then instructs the operator where to probe. The system guides the probing until the fault is found, ensuring the minimum number of probes to find the fault. The fault information is then output on the ticket printer.

The mechanical implementation of the 'probe' user by the operator is determined by the user; any type of connection to the board may be used. It is simply wired to an unused LTM/DTM pin; this pin reference is input to the system prior to the learn procedure.

This option provides a highly optimised and low overhead cost solution to the problem of digital fault location. It can also be used for the digital part of hybrid boards.

This option can only be installed on systems fitted with the Mini-Disc Option, and diagnostic programs must be stored on the disc.

A.19 Footswitch Option

This option, comprising the Footswitch Interface (FSI) installed in the Control Box, and a dual footswitch on a flying lead, permits 'hands-off' operation of the system, primarily intended for use in conjunction with the Automatic Digital Diagnostics option (ADD).

Using the ADD, it is necessary to probe internal nodes of digital boards, whilst simultaneously operating the PROCEED key. The first function of the footswitch replaces the hand operation of the PROCEED key. During the operation of the ADD, it is also necessary to answer questions with a Y (yes) or N (no), for which the VDU keyboard may be used. The dual footswitch also operates as Y or N.

In systems without a VDU but with ADD, the footswitch is essential in order to command the system to retry a probe sequence after an operator misprobe.

The FSI may also be used as an input from, for example, a microswitch on vacuum jigs to cause automatic test program initiation when the board has made contact with the 'bed of nails'.

A.20 Customer Interface Module (CIM)

Occasionally, it is necessary for customers to provide their own special interfacing electronics between the test system and the UUT. To minimise the costs involved, a CIM can be employed.

This is a PCB identical in size and connectors to a standard test module, with a prototype 'bread-board' pattern. Customers circuitry can make use of the system test rack power supplies (+15V, -15V, +5V) but, for safety, no access is permitted via software. Control of circuitry on a CIM is best performed by other standard test facilities being connected via the test jig to the CIM (e.g. LTM pin).

Use of a CIM has two distinct benefits:

- 1 In cases where the same circuit is necessary for more than one test jig, it is not necessary to replicate the circuit for each jig.
- 2 Specialised circuits are installed in the system and protected from damage.

A.21 Vacuum Adaptor for Test Platform

In cases where the number of pins on a 'bed-of-nails' (or the mechanical characteristics of the UUT) preclude the use of the built-in hold down mechanism, test platforms can be installed with this option providing vacuum pull down.

The maximum board area is 330 x 190 (13" x 7½"), and the adaptor can be supplied separately for fitting to an existing platform, or already pre-assembled to the platform.

The adaptor does not include the vacuum supply or connecting hoses, which will vary with each application. If a compressed air supply is available, the easiest way to generate a vacuum is by a venturi pump. Wayne Kerr will provide advice in the selection of a vacuum source for a particular application.

A.22 Real Time Test Option

Dynamic testing of microprocessor-based circuits is achieved with this option, which consists of a pair of test modules: the Data Compression Module (DCM) and the Clock Qualifier Module (CQM).

The DCM is essentially a parallel signature analyser and provides serial compression of 16-bit parallel words over a number of samples, resulting in a 16-bit signature which represents the activity of all 16 bits over the number of samples. Having learnt this signature initially from a known good unit, this is then stored in the test program and used to verify the operation of subsequent test units.

Associated with the 16-bit data inputs is a clock input, to instruct the DCM when to sample the data. Also provided are start and stop inputs to allow for selection of an appropriate test sequence from general activity.

The stimulus for the board under test will generally be available on the board as part of its normally installed software, and programming involves simply the selection of an appropriate sequence using the start/stop capabilities.

The CQM provides auxiliary functions, including automatic generation of the start and stop signals from pre-loaded information. 16-bit registers on the CQM are compared at real time with actual data, resulting in the generation of the relevant signal upon equivalence. This data would typically be the address bus of the microprocessor, thus the CQM would be loaded with start and stop information as software addresses.

The CQM also provides 8 clock qualifiers, for use in structured circuits, where time or positional sensitising sequences can be formulated for more precise diagnostics.

The maximum clock rate (i.e. pattern input rate) is 4 MHz and all inputs to the DCM and CQM are high impedance and thus, with a variable threshold in the range -15V to +15V, can be interfaced to all logic technologies.

Extra DCM's can be installed to allow for simultaneous testing on more than 16 bits parallel data.

The detailed electrical specifications of each module are in Appendix A.23 (DCM) and A.24 (CQM).

A.24 Clock Qualifier Module (CQM)

The Clock Qualifier Module provides automatic generation of the auxiliary input signals for the Data Compression Module. The contents of two internal 16-bit registers are compared with the input data to generate the start or stop signals on equivalence. An 8-bit register is compared with 8 clock qualifier inputs to generate valid clocks.

The start and stop signals are synchronised to the clock input to prevent false triggering of the DCM.

Inputs from UUT

- 16 Addresses
- 8 Clock Qualifiers
- 1 Clock

Control Inputs/Outputs (to DTM)

- 16 Data Lines
- 3 Address Lines
- 1 Strobe
- 1 Threshold Input (Voltage Input) - defines threshold level V_{SH} , defaults to 1.55 V if not connected

Outputs to DCM

- Valid Clock
- Valid Start
- Valid Stop

All UUT Inputs

Input Resistance	> 200 k Ω
Input Capacitance	$\approx$ 15 pF
Input Voltage Range	$\pm$ 15 V
Input Threshold Accuracy V_{SH}	$\pm$ 10% $\pm$ 100 mV
Address Set Up to Clock Edge	80 ns Minimum
Address Hold from Clock Edge	75 ns Minimum
Clock to Start/Stop O/P Delay	130 ns Maximum
Qualifier Set Up to Clock Edge	80 ns Minimum
Qualifier Hold from Clock Edge	75 ns Minimum
Clock In to Clock Out Delay	130 ns Maximum

APPENDIX B1: INSTALLATION

IMPORTANT A.C. Power input to the Control Box must be from the connector provided at the rear of the Test Console and NOT direct from a supply socket. Also, care must be taken to ensure that the 64-way connectors are inserted correctly: if excessive force is used, the polarisation key can be overcome.

A test station consists of a Control Box and a Test Console, the two being interconnected by one power lead, one 64-way and one 37-way ribbon cable. An a.c. supply is required by the Test Console, which has an isolation switch at the rear. In the down position the system becomes live, indicated by a red LED on the keypad, but is operational only when the POWER key has been pressed to light the yellow LED.

A programming station has the two units as described above, with the addition of a Visual Display Unit (VDU). This has a multiway cable connected to the front of the SIO module in the Control Box, providing the required CCITT V24 (RS232C) serial interface. The VDU requires a separate a.c. power supply connection and should be set to operate: ON LINE, FULL DUPLEX, NON-TRANSPARENT and NON-PAGE, where these facilities are not provided automatically. The interface between the SIO module and the VDU conforms to CCITT V24 (RS232C) and requires only a three-wire connection, i.e. Ground, Transmit Data and Receive Data. Handshake facilities are not available on this link (Port 1 in programming language) and the normal handshake switch of S203 bit 1, when set to '1', duplicates the ticket printer messages on the VDU. This permits a saving of paper during program development, since the printer may be switched off.

As supplied, the VDU and SIO cards are set to operate at 9600 Baud rate, 7-bit data with ONE stop bit and EVEN parity. Thus S203 on SIO 1 is normally set to FC (Hex) or 1111 1100. The COMP/PERI switches should be set as for COMPUTER (see page B-5).

The programming software operates the VDU as if it were a 'glass teletype', i.e. no assumptions are made concerning control options such as cursor control. Only the control codes, carriage return (CR), line feed (LF) and back space (BS) are used to control the display. However, there is no restriction on the use of

control characters when OUT P1 statements are written, but caution should be exercised in their use, as their effect may be specific to type of VDU. In such cases, programs transported to a different VDU may not display as expected.

APPENDIX B2: SIO MODULES - Setting-up

B2.1 Interface to ATE System

Two SIO modules may be fitted to a system and must be set up correctly to interface to the control bus. The settings required for both modules are listed below and all other sections of switches S201 and S202 can be ignored. Any page straps also may be ignored.

SWITCH SECTION		LABEL	PORT 1 (VDU)	BOTH	PORT 2 (AUX)
S202	8	CLK		1	
	7	MRQ2		0	
	6	MRQ1		0	
	5	IORQ		1	
	4	NMI		0	
	3	INT		0	
S201	5	AB7		1	
	4	AB6	1		0
	3	AB5		1	
	2	AB4		0	
	1	AB3		0	

Note: For switches S202 and S201, '1' state is towards front panel.

B2.2 Interface to RS232

S203 settings Settings of switch S203 (on the SIO module) inform the system of the data parameters of the RS232 link. These settings may not be common with other Wayne Kerr instruments using this module. With the exception of data bit 1, they are common to both Port 1 and Port 2.

DATA BITS	SWITCH SECTION	SETTINGS	FUNCTION
DB7,6,5	8,7,6	111000 001 010 011 x100 101 110 111	110 150 300 600 1200 2400 4800 9600 Baud Rate
DB4	5	10 11	Odd Parity Even parity
DB3	4	10 11	No parity bit Parity bit as DB4
DB2	3	10 11	One Stop bit Two Stop bits
DB1	2	10 11	No handshake Port 2 No printer mimic Port 1 Handshake on CTS etc (2) Mimic printer on VDU (1)
DB0	1	00 1	Seven bit Data Eight bit Data

Note: For switch S203, '0' state is towards front panel.

COMP/PERI switches Normally, SIO ports will be used to drive printers or VDUs and the ATE will be the COMPuter. In these cases, viewing the module in its normal position in the rack, set the top four switches to PERI and the bottom two to COMP.

Where the module connects to another computer, these switches should all be set the other way, i.e. top four to COMP and bottom two to PERI.

The basic rule is that if the other equipment is to be treated like a peripheral, use the first settings; if the other equipment is to be considered as master, use the second settings.

Connector wiring When set as COMPuter, the SIO module transmits data on pin 3 and receives it on pin 2. In RS232 terms, it looks like the Data Communication Equipment (DCE). Minimum wiring between SIO port and peripheral/computer for a non-handshake connection is:

2-2, 3-3, 7-7

Pin 1 (protective ground) may be wired to the plug shell.

For full wiring see section on Port 2 anomalies.

Where a Facit VDU is supplied, the necessary handshake connections are wired in the cable provided.

B2.3 Port 2 (Auxiliary SIO) - specific features

End-of-line Character All IN P2 program statements require <RETURN> (ASCII 0D) to signal completion of input data. OUT P2 statements output only those characters listed in the statement. Unlike OUT P1 statements, there is no automatic CRLF (ASCII 0D 0A) to start a new line and these must be inserted by the programmer if required. They are omitted to allow protocol communication, with some types of peripherals, which parasitic characters would veto.

Program Transfer All test Programs may be transmitted in total, or in part, through Port 2 either with or without hardware handshake. See section 8.1, under 'O' at the FUNCTION level. This is primarily to allow the program to be printed externally.

A functional program may be down-loaded from a word-processor or another ATE,

using the RECEIVE command at program selection time. The transfer is controlled by a software handshake using ETX and STX characters and is described fully in section 8.1. In addition to the software handshake the hardware handshake may be used.

The Transmit command 'T', when at FUNCTION level of the TPG, is the complement of RECEIVE and supports the software protocol requirement for transfer of programs as described in section 8.1. Note that the COMP/PERI settings must be complementary, not similar, when transferring between ATE's.

Hardware Handshake If S203 DB1 (switch marked 2) is set to 1, the Auxiliary SIO operates two levels of hardware handshake:

- 1 Verification of attachment to working peripheral (or computer). When used as COMPUTER, i.e. COMP/PERI switches are set with top four to PERI, lower two to COMP:

ATE asserts DSR on pin 6

ATE expects DTR on pin 20

When used as PERIpheral, these pins are reversed.

- 2 Verification that it is clear to send or ready to receive data. When used as COMPUTER:

ATE asserts CTS on pin 5 to show data may be sent

ATE expects RTS on pin 4 to confirm it may send data

When used as PERIpheral, these pins are reversed.

Connector wiring Normal wiring between SIO port and peripheral/computer is by the following connections:

2-2, 3-3, 4-4, 5-5, 6-6, 7-7, 20-20

Some devices use alternative handshakes to inhibit data transmission, e.g. pin 8 RLSD instead of DTR, and in these cases the wiring should be adjusted to suit. In all cases the equipment manual should be consulted to determine the need for - and method of - handshaking.

APPENDIX C: ABBREVIATIONS GLOSSARY

ACM	Analog Command Module
ADD	Automatic Digital Diagnostics
ALT	Alternate
APG	Analog Power Ground
APP	Apply
ASG	Analog Signal Ground
ATM	Analog Test Module
bs	Backspace (ASCII code '08' H)
BS	Block Switch
BSM	Block Switch Memory
CAL	Calibration
CCM	Clock/Calendar Module
CIM	Customer Interface Module
CPG	Continuity Program Generator
CPS	Console Power Supplies
CPU	Central Processing Unit
CTA	Counter Timer line A
CTB	Counter Timer line B
<CTRL Q>	Simultaneous operation of CTRL and Q keys
DGND	Digital Ground
DMA	Direct Memory Access
DTM	Digital Test Module
DVA	Drive Voltage Anode
DVK	Drive Voltage Cathode
EOS	End of Sequence
EOT	ASCII code '04' H
EPROM	Erasable Programmable Read Only Memory
ERR	Error
<esc>	Escape (return to the beginning of a statement)
ETX	ASCII code '03' H
FPG	Functional Program Generator
FSI	Foot Switch Interface
GTO	Gate Time Output
HDR	High Density Ram
IEEE	IEEE Control Interface
IPG	Interactive Program Generator
ITM	Impedance Test Module
LED	Light Emitting Diode

lsb	Least Significant Bit
LTM	Logic Test Module
MP	Maths Pack
msb	Most Significant Bit
MV	Measured Value
PCM	Power Control Module
PDM	Power Distribution Module
PEM	Prom Enable Module
PGM	Pulse Generator Module
PLM	Program Link Module
PSG	Pseudo Signal Ground
PSM	Power Supply Module
PSU	Power Supply Unit
PTB	Power Transition Board
RAM	Random Access Memory
REM	Remove
<ret>	Carriage Return (ASCII code '0D' H)
RRM	Reed Relay Module
RTM	Real Time Module
SIO	Serial Interface (also used as Serial I/O Port)
SLET	String Variable Store input command
SOM	Shorts & Opens Module
sp	Space (ASCII code '20' H)
SQRT	Square Root
SSM	Sinewave Source Module
STX	ASCII code '02' H
TBI	Test-Bus Interface
TIC	Test Interface Connector
TIM	Timer Interface Module
TMH	Test Module Handler
TP	Test Program
TPG	Test Program Generator
TPM	Test Program Module
TRC	Test Rack Controller
UUT	Unit Under Test
VDU	Visual Display Unit
VIM	Voltage/Current Source Module
VREF	Voltage Reference
ZTM	Tristate Impedance Test Module x3

APPENDIX D: DIRECT POLLING OF VDU

In exceptional circumstances, the programmer may wish to interrupt a program without using an "IN P1" statement, which stops the program and demands a key stroke. Special port manipulation allows direct monitoring of the Status and Data registers of the SIO cards. The following set of statements will detect that a key has been pressed since the last time the keyboard was read and will then put that key character in a string store:

```
IN,P229,NØ           % get status byte (fills the string with 80 copies of it)
SCOMP,NØ(Ø), '60'     % compare with NO DATA READY
GOIF GO,S1000         % if No key pressed, skip program
IN,P224,NØ           % get Data byte whether wanted or not, this clears status byte
```

```
SCOMP NØ(Ø),"A"       % If "A" pressed go to
GOIF GO S2000         % Routine appropriate to "A"
SCOMP NØ(Ø) "B"       % If "B" pressed go to
GOIF GO S3000         % Routine appropriate to "B"
```

} Optional
further
vectoring

S1000

For Port 2 Status byte is P165
 Data byte is P160

An extension of this technique will allow the operator a few seconds to respond but will default if no key pressed:

```
S1000 OUT,P1, "YOU HAVE 3 SECONDS TO REDIRECT PROGRAM"
S1010 LET,M1=0
```

```
S1020 IN,P229,NØ       % Status register
      SCOMP,NØ(Ø), '60' % Status byte compare
      GOIF,NOGO,S2000   % Key pressed, so jump
      LET M1=M1+1       % Loop counter
      COMP,M1,HL30       % This number determines time
      GOIF,GO,S1020     % Try again
```

S1100 Default program % No key pressed in 3 seconds

S2000 IN,P224,NØ % Key pressed, clear data byte

etc

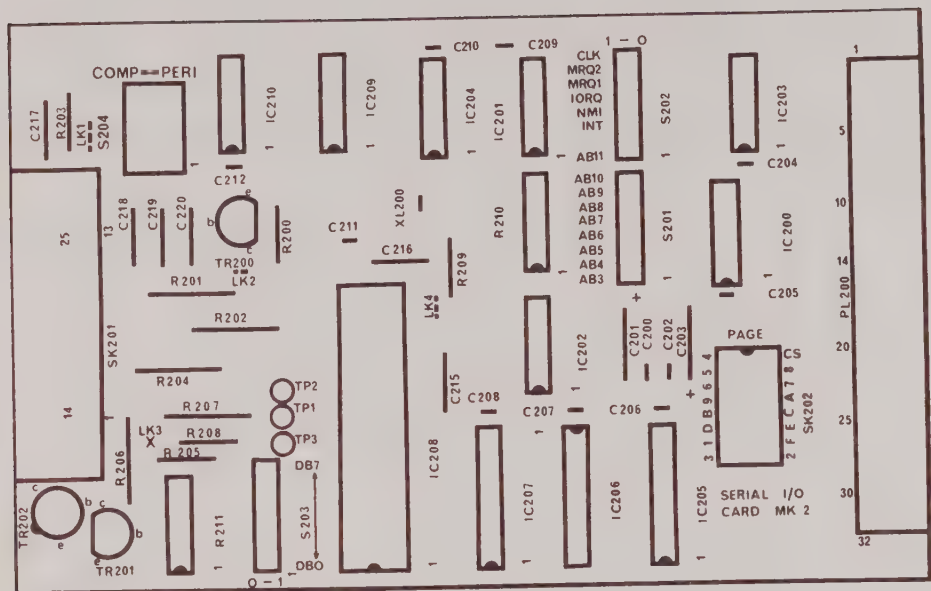
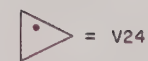


Fig. D.1 SIO Card Layout

I.C.	+5V	GND	2V	-12V
200	14	7		
201	14	7		
202	14	7		
203	14	7		
204	14	7		
209		7	14	1
210	14	7		

NOTE

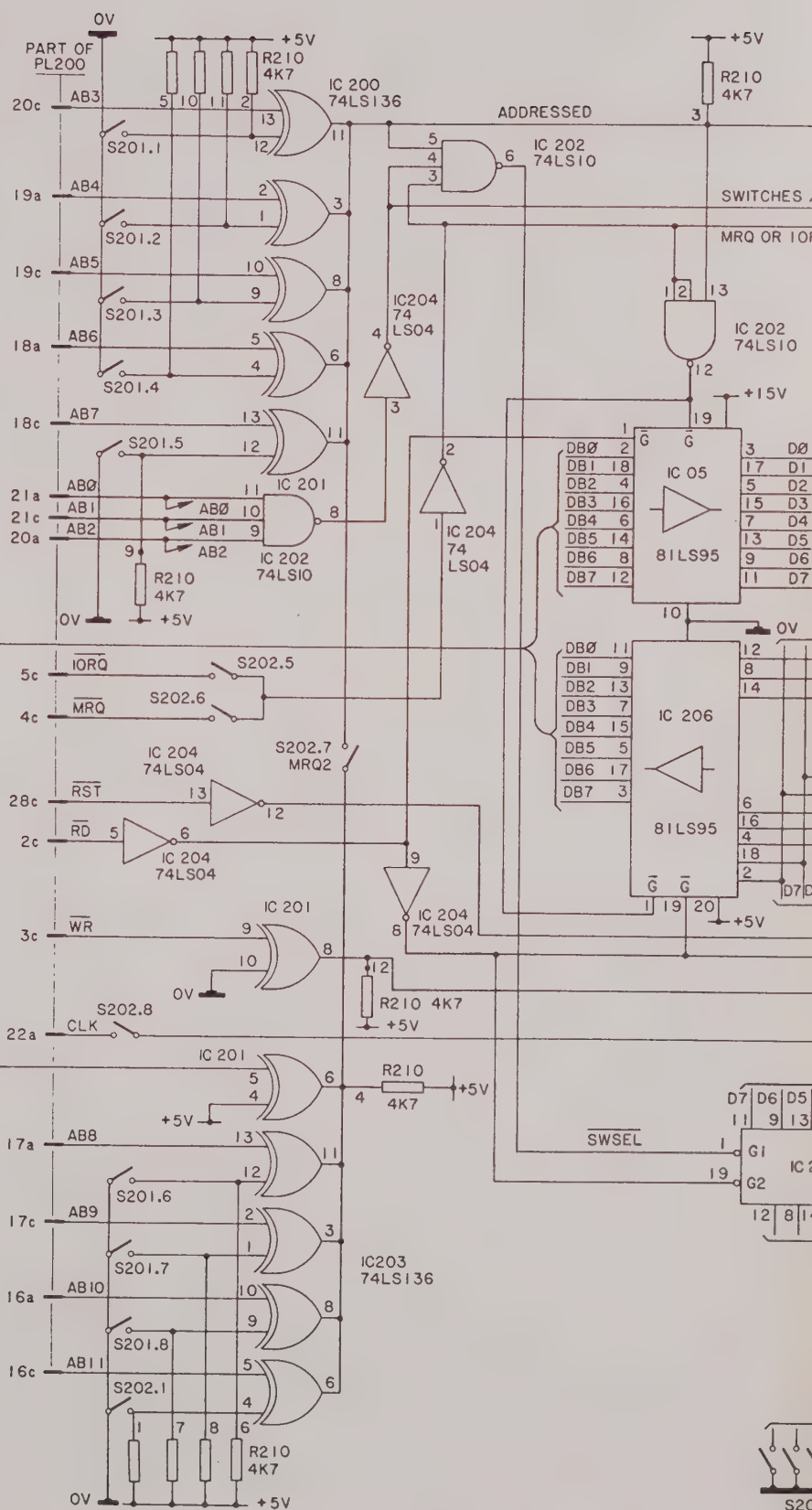
R 201, R 202, R 204,
R 206 AND R 207
ARE NOT FITTED
TO STANDARD CARD.
LINKS TO PINS (9) AND (25)
ARE ALSO OMITTED.



PART OF
PL200

32a	DB0
32c	DB1
31a	DB2
31c	DB3
30a	DB4
30c	DB5
29a	DB6
29c	DB7

6c	EN1	7
6a	EN2	9
7c	EN3	8
7a	EN4	1
8c	EN5	2
8a	EN6	4
9c	EN7	3
9a	EN8	14
10c	EN9	15
10a	ENa	13
11c	ENC	5
11a	ENE	12
12c	END	6
12a	ENF	11
13c	ENF	10



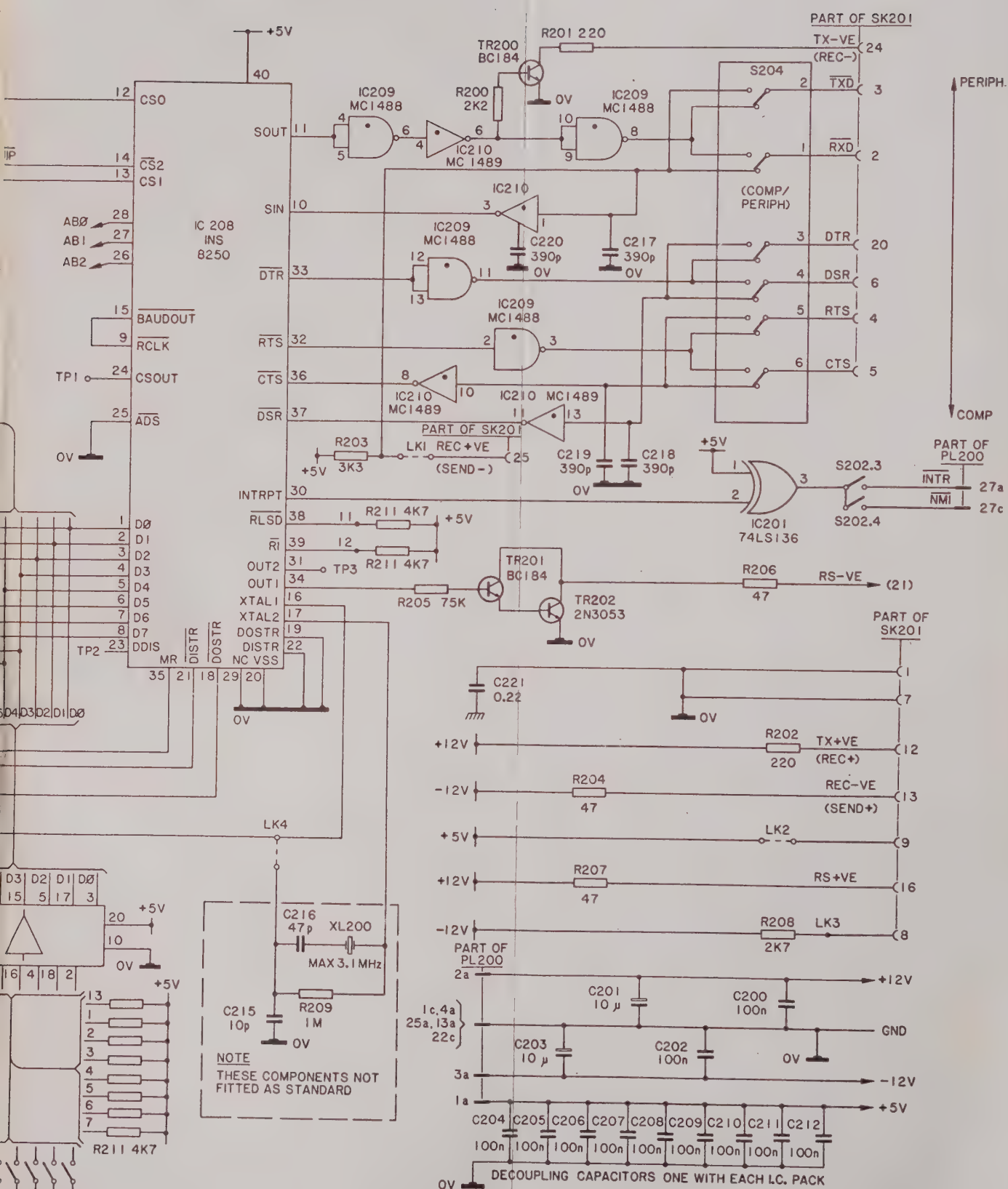
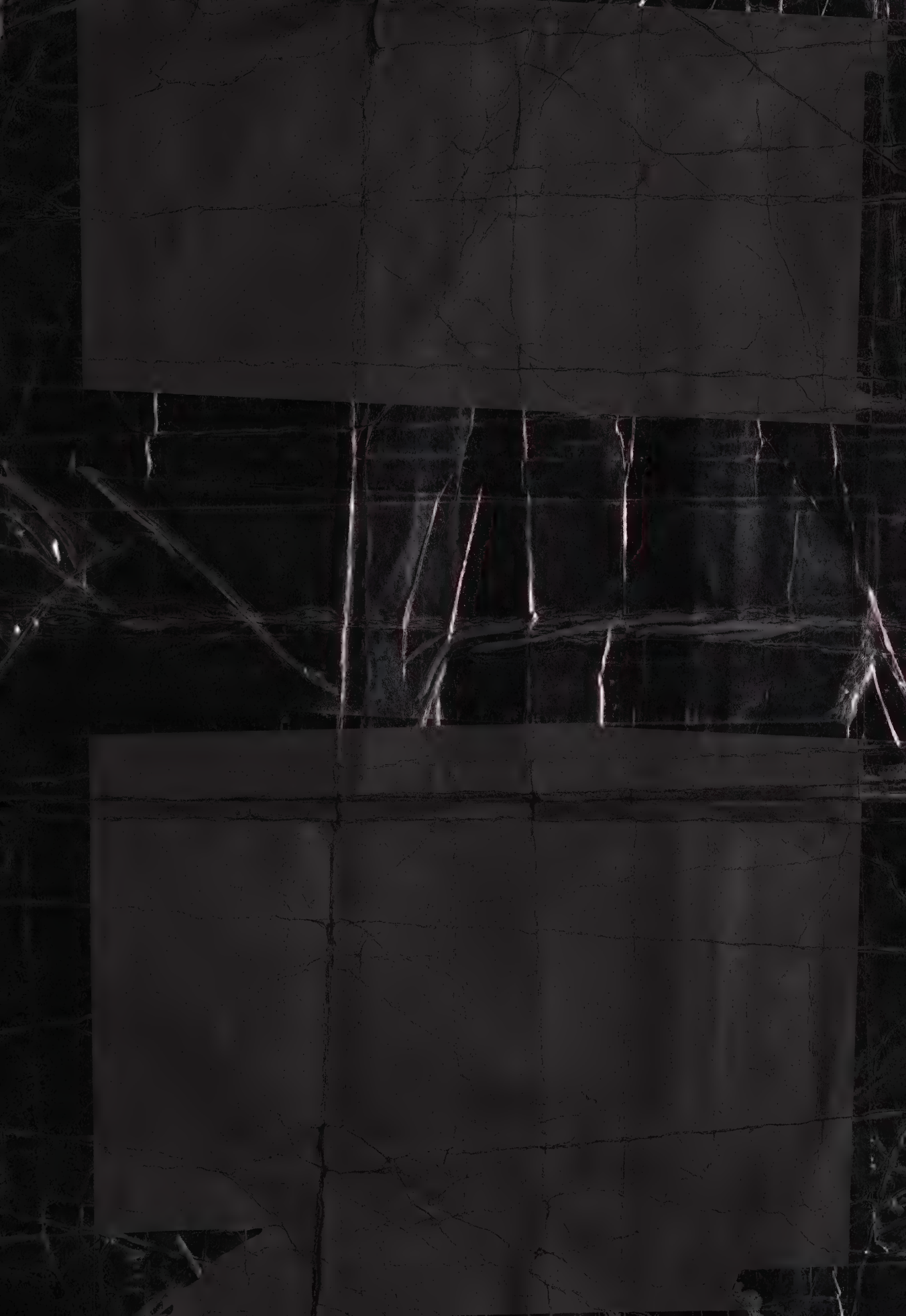
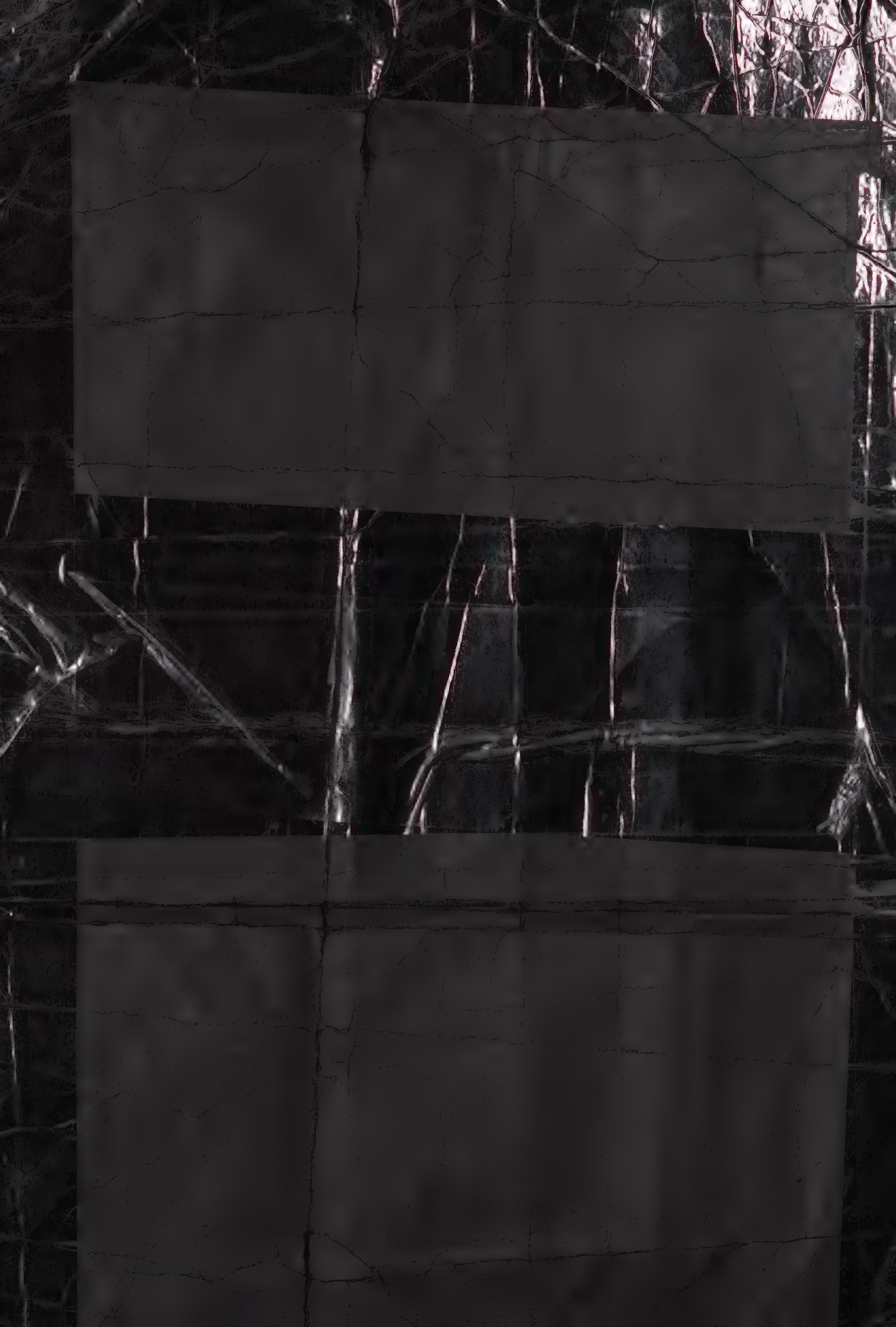


Fig. D.2 SIO Circuit Diagram





Supplement to TP218

WAYNE KERR

REFERENCE MANUAL FOR WAYNE KERR
ATE AUTOMATIC DIGITAL DIAGNOSTIC FACILITY

The Reference Number of this Manual is TP 217

CONTENTS

	PAGE
1 INTRODUCTION	4
2 TEST PROGRAM GENERATION	5
2.1 Test Program Quality	6
2.2 Driver/Sensor Probe	7
3 GUIDED PROBE LEARNING MODE	7
3.1 'L' - Learn Signatures Command	9
3.2 'F' - Fill Command	11
3.3 'U' - Unconnected Device Command	13
3.4 'K' - Kill Device Command	13
3.5 'C' - Connect Pins Command	14
3.5.1 Editing function	14
3.5.2 Non-probeable device pins	15
3.6 'V' - Verify Commands	16
3.6.1 'M' - Manual Verify	17
3.6.2 'V' - Verify Auto	18
3.7 'X' - Exit Command	18
3.8 'Z' - Delete All Command	19
3.9 'R' - Reassign Probe Commands	19
3.10 'P' and 'Q' - Device Data Listing Commands	19
3.10.1 Outputs Group	21
3.10.2 Input Group	21
3.11 'S' - Strip Connections with Driver/Sensor	22
4 GUIDED PROBE DIAGNOSTIC MODE	22
4.1 Operator Misprobes	23
4.2 Feedback and Diagnostic Resolution	26
4.3 Bidirectional/Bus Structured Devices	28

ILLUSTRATIONS

Figure 1 - Test Program Structure	6
Figure 2 - Example Circuit	10
Figure 3 - Defining Dummy Input Pin	15
Figure 4 - 'P' & 'Q' Device Data Listing	20
Figure 5 - Example for Operator Misprobe	24
Figure 6 - Example Circuit with Feedback	27
Figure 7 - Feedback Inherent on Bus Structured Devices	28

1 INTRODUCTION

Use of the guided probe facilities involves two main stages:

Firstly, the writing of a GO/NOGO test program with respect to the external connections wired to the Driver/Sensor pins of the appropriate Test Modules.

Secondly, the learning and verification of the signatures of the internal nodes of the whole board under test.

This second stage adds circuit connectivity and signature information to the test program and is used to put the tester automatically into the guided probe diagnostic mode when the test program is run and a fault is encountered in the diagnostic part.

2 TEST PROGRAM GENERATION

Test program generation is structured into the following three consecutive stages (Also see Figure 1):

- i Power supply set up and pre-initialisation pin state testing
- ii Initialisation
- iii Signature collection (see Figure 1)

- i Power supplies must be allowed sufficient time to settle and, checks for setting up conditions should be made prior to initialisation.
- ii Since the signature collection technique is to loop the test program several times to check for consistency, it is required, under fault-free conditions, that the logic level of each node is completely predictable. It is essential that the test program input statements initialise every logic node in the circuit to a known logic state. The start of the initialisation process is defined in the test program by the "DIAGBEGIN" statement.
- iii After initialisation the start of the signature collection process is defined in the test program by the "SIGON" statement. The end of this stage is defined by the "DIAGEND" statement, and it is between the first "SIGON" and the "DIAGEND" statement that every node of the board under test should be thoroughly exercised and tested to ensure non-trivial and unique signatures.

During this stage it is possible to turn off signature collection by using the "SIGOFF" and "SIGON" statements in pairs as shown in Figure 1. These statements are necessary when conditional branching is used to set a device into a known state before continuation of the main test program flow.

An example of "SIGOFF" use in suspending signature accumulation would be when an operation needs to be performed a variable number of times (such as reading INTEL 8273 status registers): the number of tests will vary and would result in a varying signature if collection were still switched-on.

Conditional branching outside the "SIGOFF-SIGON" boundaries should not be allowed as inconsistent signatures may result.

N.B. Print-out statements should not be used between the "DIAGBEGIN" and "DIAGEND" statements as these will impede signature collection, since every time the program is run during guided probing the statements will be printed out.

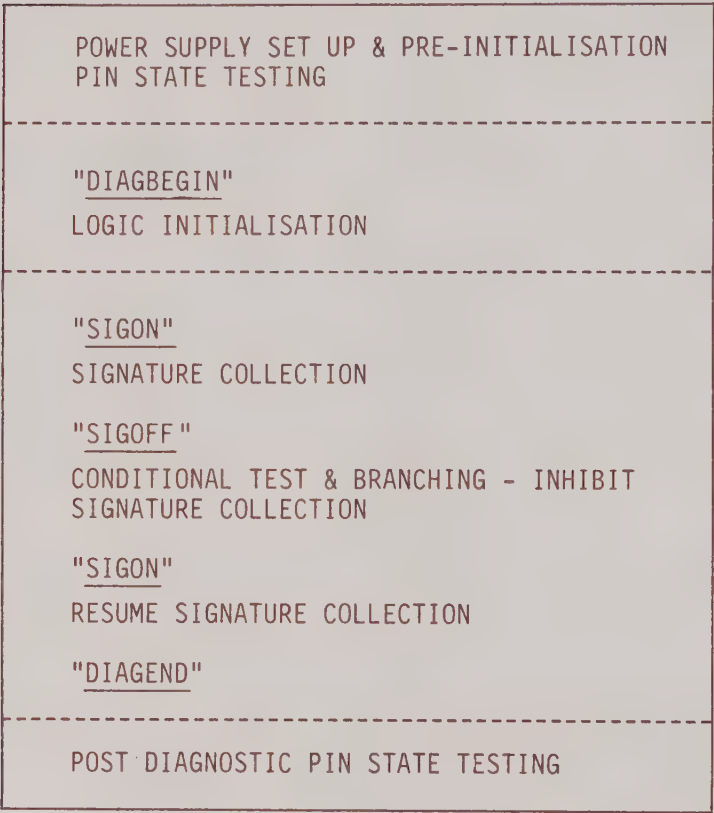


FIGURE 1 - TEST PROGRAM STRUCTURE

2.1 Test Program Quality

The Test Program must fully exercise every node on the board so that all possible stuck-at faults (and bridging faults) on gate inputs and outputs will be detected at the edge connector to drive/sensor interface.

LSI devices should, in addition, be given very thorough functional tests to exercise fully their multiple modes of operation.

Observance of the above will ensure a test program of high quality resulting in:

- a Non-trivial signatures (i.e. not stuck-at-one or zero).
- b *Almost totally unique signatures at every circuit node.

Non-trivial signatures are essential for the guided probe algorithm to work in probing all faulty nodes.

Unique signatures, whilst not essential, do make the signature learning process easier since no incorrect connectivity questions will be asked of the user. Non-unique signatures may also be an indication that not all stuck-at or functional faults have been tested for.

- * Unique signatures cannot be achieved when single input, single output non-inverting buffers are used, nor where two inverters are cascaded.

2.2 Driver/Sensor Probe

When designing the test program, one drive/sensor pin should be left unconnected to the circuit under test. This spare driver/sensor pin will be assigned as the guided probe later on in the guided-probe learning mode.

3 GUIDED PROBE LEARNING MODE

When a high quality test program has been written, run and checked for correct operation, the guided probe learning mode is entered by operation of the 'G' key at the terminal keyboard.

On entry into guided probe learning mode the following message will be displayed

```
ADD      V1.7
04FE    FREE BYTES
```

The number of free bytes is a hexadecimal indication of the free space available to the user. If at any time the user wishes to know how much free space is available he simply exits and then re-enters Guided probe.

The significance of the free space is that the user must allow 2 bytes per D/S pin (any LTM or DTM pins) which will be connected when using the Fill command.

If at any time a potential overflow of allowable space is encountered, the message

```
O/FL
```

will be output. If this occurs during a Fill or before Fill is executed, sufficient space must be deleted from the test program. Until a Fill

is successfully completed, the learnt ADD data is not very useful.

The user is requested to assign a driver/sensor pin as the signal input for the guided probe when entering the learning mode for the first time with the current test program. If the user has currently loaded a test program with previously learned device data, the driver/sensor last defined before exiting from the guided-probe learning mode is displayed as the probe. The test program is then executed up to the DIAGBEGIN statement.

It should be noted that from this point onwards during the learning sequence the board is continually under power (assuming that the power-on statements occur prior to DIAGBEGIN) until leaving the guided-probe mode (via the 'X' command), therefore care should be taken to avoid damage to the board.

The system next enters the guided-probe command level with the prompt:

COMMAND ?

Any of the available commands may now be activated by operating the single key for the command. These commands are explained below in a typical sequence that would normally be followed when entering the guided-probe learning mode for the first time. It is essential that the tester is loaded with a known good board, when entering the guided-probe learning mode, or signature and connectivity errors will occur. Otherwise the system will not permit learning of signatures from a failing board.

Command list summary

C	Connect
F	Fill
K	Kill
L	Learn
M	Manual
P	Print device data
Q	List device data
R	Re-assign probe
S	Strip
U	Unconnected list
V	Verify
X	eXit
Z	Delete all

3.1 'L' Learn Signatures Command

This command is activated by the 'L' key.

The process of learning signatures has two objectives: to learn what the signatures are at various points on the board and to discover the topology (connectivity) of the circuit.

When learning signatures, the user specifies a device and output pin (or pins). This pin is then probed and the signature measured. If the signature is the same as the signature measured elsewhere on the board, the user is asked whether or not the pins are connected. If the reply is Yes this adds to the topological data.

Having measured the outputs on a device, the inputs are then specified and the signatures on the input pin measured. As each signature is measured, the check is made to see if the signature has been met before and if so the user is asked whether or not there is a connection between the points. Each Yes answer gives more topological data.

It should be very rare for a No answer to be given to the questions about connectivity, as distinct nodes with the same signature should not normally occur (See Test Program Quality above). One major exception is when non-inverting buffers are used.

The example of a signature learning session for a small circuit will show the main ideas, see Figure 2.

```

L
LEARN
DEVICE - IC3
O/P PIN - 2
PROBE IC3 PIN 2 (Place probe and operate 'PROCEED'
                  button or footswitch if fitted)
1A6F
O/P PIN - 3
PROBE IC3 PIN 3
3F9A
O/P PIN (Terminate this sequence with <Ret> only)
I/P PIN - 1
PROBE IC3 PIN 1
02F4
I/P PIN - 4
PROBE IC3 PIN 4
2763

```

I/P PIN - 9
PROBE IC3 PIN 9
69CD
I/P PIN - (Terminate with <Ret>)
LEARN
DEVICE - IC2
O/P PIN - 3
PROBE IC2 PIN 3
2763
IS IC3 PIN 4 JOINED
Y
O/P PIN - (Terminate with <Ret>)
I/P PIN - 1
PROBE IC2 PIN 1
02F4
I/P PIN - 2
PROBE IC2 PIN 2
37B1
I/P PIN - (Terminate with <Ret>)
LEARN
DEVICE - IC1
O/P PIN - 3
PROBE IC1 PIN 3

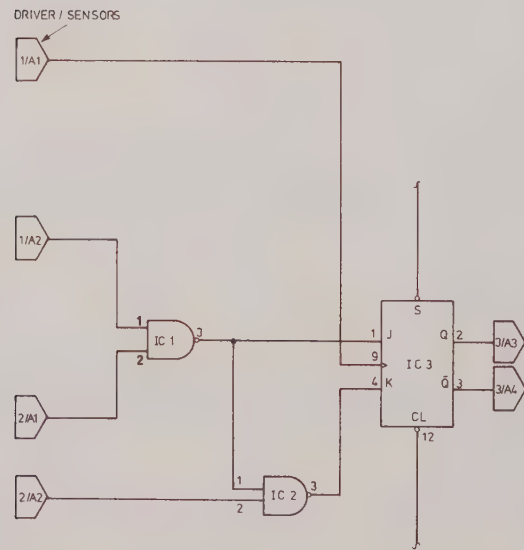


FIGURE 2 - EXAMPLE CIRCUIT

```

02F4
IS IC2 PIN 1  JOINED
Y
IS IC3 PIN 1  JOINED
Y
O/P PIN - (Terminate with <Ret>)
I/P PIN - 1
PROBE IC1 PIN 1
2345
I/P PIN - 2
PROBE IC1 PIN 2
789A
I/P PIN - (Terminate with <Ret>)
LEARN
DEVICE - (<Ret> on Device returns to guided probe command level)
COMMAND ?

```

The circuit gates may be learnt in any sequence but outputs must be defined and learnt before specifying the inputs controlling those outputs.

A check is made for consistent probe contact when the 'PROCEED' key is operated. The system runs the test program several times for signature corroboration. If the signatures are not consistent, a 'REPROBE' message is displayed.

Device names may contain up to six alpha-numeric characters. Spaces may not be used in the name. Pin definitions may be in the range 1 to 127 or A to Z. Any other values cause a number error to be flagged. Up to 255 device names may be used. For multi-element devices, each element is learnt separately but, assuming a common device name is used, this does not affect the maximum number of device names.

3.2 'F' Fill in device data with driver/sensors

This command is activated by the 'F' key and is used to learn the signatures on all driver/sensors and to discover to which device pins they are connected. Signatures are collected automatically from all driver/sensor pins. For each signature, the device data is scanned for a signature match.

When a match is found, the user is queried as to whether or not the device pin found is connected to the matching driver/sensor. If the reply is Yes this connectivity information is added into the device data structure.

As in the case of the LEARN command it should be rare to answer No to the connectivity questions asked of the user so long as the test program is of high quality and virtually unique signatures are generated.

An exception to this arises when a learnt node signature is an unavoidable stuck-at-one or zero, as in the case of input pull-up or pull-down resistors, and also if some of the defined driver/sensor pins are not used and will also have trivial signatures learnt from them. Here it is possible that irrelevant connectivity queries will be made and so that the user is prompted along with the message "WARNING D/S STUCK" to make the user aware that the reply is almost certainly No since no used driver/sensor should collect a trivial signature.

The system may be commanded to learn further devices after a FILL command and then execute further FILL commands to connect in extra driver/sensors.

```
Example FILL sequence:      F
                             FILL
                             IS IC3 PIN 9 JOINED
                             TO D/S - 1/A1
                             Y
                             IS IC1 PIN 1 JOINED
                             TO D/S - 1/A2
                             Y
                             IS IC1 PIN 2 JOINED
                             TO D/S - 2/A1
                             Y
                             IS IC2 PIN 2 JOINED
                             TO D/S - 2/A2
                             Y
                             IS IC3 PIN 2 JOINED
                             TO D/S - 3/A3
                             Y
                             IS IC3 PIN 3 JOINED
                             TO D/S - 3/A4
                             Y
                             N.C PIN(S)
                             NONE
                             COMMAND ?
```

At the end of the connectivity queries, the FILL command scans the device data structure for any unconnected device pins which it lists out before

returning to the guided probe mode command level.

3.3 'U' List unconnected device pins

This command is activated by the 'U' key and performs the same function of listing unconnected device pins (input and output) as at the end of the FILL command.

The 'U' command may be used at any point during the learning of device data and is particularly useful to determine which device pins have been disconnected while excluding a device from the data structure after a KILL command.

```
Example 'U' sequence:      U
                           N.C. PIN(S)
                           IC5 PIN 12
                           IC10 PIN 15
                           COMMAND ?
```

3.4 'K' Kill previously learned device

This command is activated by the 'K' key. It is available to the user for editing the learned data when an error has been made during one of the data learning commands i.e. LEARN, FILL or CONNECT and recovery is not easily achieved by re-using the command.

The KILL command deletes from the data structure all the elements of the named device. For example if killing a quad NAND device then all 4 elements under the common device name would be removed.

Killing a device disconnects that device's inputs and outputs from other learned devices and leaves them unconnected such that when the killed device is re-learned (hopefully without error this time) then "is this device joined?" queries will again be asked of the unconnected pins so that they will be rejoined within the device data structure.

There is no limit to the number of times a device may be killed and re-learned. However, if all devices are to be killed, the 'Z' command (see 3.8) should be used instead of killing all devices individually, as this is more efficient in data storage.

When killing a device that drives a wired-AND connection, then all devices with inputs or outputs connected to that wired-AND point are unconnected but, of course, are reconnected when the killed device is

re-learned and all the "is this device joined?" queries are raised again.

Device pins which are left unconnected after a kill command may be listed out by use of the 'U' command (see section 3.3).

```
Example KILL sequence:      K
                             KILL
                             DEVICE - IC22
                             COMMAND ?
```

3.5 'C' Connect device pins command

This command is activated by the 'C' key and has two major uses in adding connectivity into the device data structure under user control. These are editing device pin connections and connecting in dummy non-probeable device pins.

3.5.1 Editing function

CONNECT is used to make connectivity between device pins when a No response may have been given in error to an "IS THIS PIN JOINED" query and the pin left unconnected. This command allows this situation to be easily remedied without recourse to the more drastic KILL command. Also it can be used to reconnect devices that may have been left unconnected after a KILL command and were not automatically reconnected when the killed device was re-learned.

```
Example CONNECT sequence:   C
                             CONNECT
                             DEVICE - Q1
                             O/P PIN - 5
                             DEVICE - Q2
                             PIN - 6
```

Signatures must be the same before connect is allowed

```
DIFF SIGS
DEVICE - Q1
O/P PIN - A
DEVICE - Q2
PIN - 6
```

Successful connection made

```
CONNECT
DEVICE - (terminate connect with <Ret>)
COMMAND ?
```

When using the CONNECT command as shown in the above the following restrictions apply:

- a The first device pin specified must be an output
- b The second device pin specified may be input or output
- c Signatures must be either equal or one of them unknown as in the case of a defined dummy non-probeable pin
- d A previously connected input may not be specified, instead its connected driver should be given
- e If both pins to be connected are outputs then only one is allowed to be already connected to a WIRED-AND node.

3.5.2 Non-probeable device pins

If a device pin has been learned as non-probeable then it will need to be connected into the device data under user control. The non-probeable pin facility is available to cut down probing times. This allows dummy device inputs to be specified which will never actually be probed. Instead the guided probe algorithm will look at the driving pin and, if the signature there is faulty, will continue to search from that point. For example, as in Figure 3, where a fault on IC1 goes through many levels before being visible, it is useful to force the probe to search back in the circuit as soon as possible. This can be achieved by defining an unprobeable input on IC4 and saying that it is connected to IC1.

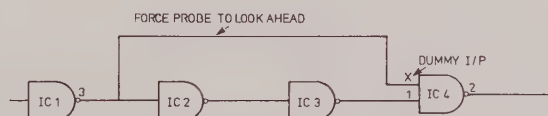


FIGURE 3 - DEFINING DUMMY I/P PIN

The learning sequence would be:

Example Learning sequence

Dummy input

L

LEARN

DEVICE - IC4

O/P PIN - 2

PROBE IC4 PIN 2

1234

O/P PIN - (Terminate output pin sequence)

```

I/P PIN - X - (Define dummy pin) or any
                character with the addition
                of "-"
I/P PIN - 1
PROBE IC4 PIN 1
2345
INPUT PIN -
LEARN
DEVICE

```

Example Connect sequence for learnt dummy pin

```

C
CONNECT
DEVICE - IC1
O/P PIN - 3
DEVICE - IC4
PIN - X
CONNECT
DEVICE -
COMMAND ?

```

3.6 Verify Commands

These are used to check the learned signatures, for verification of the device data or for manual diagnosis. There are two verify commands:

Manual - where the operator enters device and pin numbers explicitly for each probe

Verify - where the system automatically sequences through devices and pins, starting automatically or from an operator defined start point.

After having learnt signatures from a good board, it is important that these signatures are verified BY A DIFFERENT OPERATOR (to minimise consistent operator errors). The Verify command is most suitable for this.

The Manual command is most useful for manual guided diagnosis, when the operator chooses not to use the automatic fault finding probe sequence (see Section 4).

Other uses for the Verify commands are:

- i correction of connectivity due to incorrect probing in learn mode.
After correction of the signature, the relevant pins can be connected using the Connect command.
- ii when a test program is modified, the Verify commands can be used to update the signatures without having to re-learn the connectivity.

Both Verify commands will report if the program is failing, but will also report the signature - and whether it is good or bad - to allow the use of these commands in manual diagnosis of a failing board.

In the case of a signature differing from the stored one, the operator will be presented with the opportunity to replace the stored signature by the new one.

3.6.1 'M' Manual Verify Command

This command is activated by the 'M' key and will request device and pin identifiers.

Example Verify sequence:	M	
	MANUAL VERIFY	
	DEVICE - Q1	
	PIN - 1	
	PROBE Q1 PIN 1	
	FF34	(signature)
	OK!	
	PIN - 2	
	PROBE Q1 PIN 2	
	0F4A	
Signature Not Correct	LEARNED SIG - 1234	
	REPLACE ?	
	Y	
	PIN - 3	
	PROBE Q1 PIN 3	
	OK!	
	PIN -	
On to next device	DEVICE	

It is only possible to verify 'real' pins.

3.6.2 'V' Verify Auto

This command is activated by the 'V' key and gives the option to verify the input pins on all devices starting at the first device in the data structure or starting from a specified device.

Example:

```
V
VERIFY
ALL ?
Y
PROBE Q1 PIN 1
OK !
PROBE Q1 PIN 2
ETC.,
```

or

```
V
VERIFY
ALL ?
N
DEVICE - Q2
PROBE Q2 PIN 1
OK !
PROBE Q2 PIN 2
ETC.
```

3.7 'X' Exit from the guided-probe learning mode

This command is activated by the 'X' key and is used to exit from the guided probe learning mode when the device data for the test program has been created. Normally upon leaving the test program with its appended device data, it would be stored away on disc for later running in the diagnostic mode. It should also be considered good practice to exit regularly whilst learning device data and storing away intermediate stages of the program so that any catastrophic errors causing a loss of data (accidental use of a KILL or Z command) will not require a total start from scratch.

3.8 'Z' Delete whole device data

This command is activated from the command level by the 'Z' key and deletes all the previously learned device data. It should be used instead of the KILL command for deleting all data as it is more efficient in releasing data storage area.

The user is asked to confirm the destruction of the data base by the question:

KILL ALL ?

Answering 'Y' will cause the deletion to occur.

Answering anything else will cause reversion to the command level.

3.9 'R' Reassign driver/sensor as probe

This command is activated from the command level by the 'R' key and has a dual function:

- a When REASSIGN is activated it displays the driver/sensor module and pin number currently assigned as the guided probe for signature collection and diagnosis.
- b The user is queried as to whether the probe is to be reassigned to a different driver/sensor, as in the case of the test program being expanded when more modules are required to be added.

Example REASSIGN command sequence:

```
R
PROBE D/S = 7/A9
REASSIGN ?
Y
ASSIGN PROBE
7/A6
COMMAND ?
```

3.10 'P' and 'Q' - List Learned device data commands

The 'Q' command allows the user to list to the terminal or printer the currently held data on a named device. The format is as shown in Figure 4. Each element under a device name heading is grouped firstly into the output list followed by the elements input list. Every named device is also given an internal reference number.

P

DEV-NAME = A1			DEV-NO= 01			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X	X		4444		
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
02	X	X		0103		
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
03	X	X		1111		
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
04	X	X		0204		

DEV-NAME = A2			DEV-NO= 02			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X	X	X	5555		
PIN-NO	SIG	DRVR	J. N.P.	SG/DRVR	D/S	INPUT(S)
02	X	X		0303		
03	X			5511	06A6	
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
04	X	X		3646		
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
05	X	X		0A01		

DEV-NAME= A3			DEV-NO= 03			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X	X	X	5555		
PIN-NO	SIG	DRVR	J. N.P	SG/DRVR	D/S	INPUT(S)
02	X	X		0C01		
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
03	X	X		2222	06A5	
PIN-NO	SIG	DRVR	J. N.P	SG/DRVR	D/S	INPUT(S)
04	X	X		0C01		
05	X	X		0B01		

DEV-NAME= *			DEV-NO= 04			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X	X		5555		
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
02	X	X		0301		
03	X	X		0201		
00	X	X		0701		

DEV-NAME= A4			DEV-NO= 05			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X			6666	06A1	
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
02	X	X		0101		

DEV-NAME= A5			DEV-NO= 06			
PIN-NO	SIG	JOIN	W.DRVR	SIGNTUR	D/S	OUTPUT(S)
01	X	X		7777		
PIN-NO	SIG	DRVR	J N.P	SG/DRVR	D/S	INPUT(S)
02	X	X		0101		

* DUMMY WIRE-AND DEVICE

FIGURE 4 - 'P' & 'Q' DEVICE DATA LISTING

3.10.1 Outputs group

- a 'PIN-NO' column indicates the pin in hexadecimal or ASCII if alphabetic.
- b 'SIG' column indicates whether the signature is known.
- c 'JOIN' column indicates if this output drives an input.
- d 'W'.DRVR' column indicates if the output drives a WIRE-AND point.
- e 'SIGNTUR' column records the 16 bit signature in hexadecimal.
- f 'D/S' column records the connected driver/sensor module and pin number.

3.10.2 Input Group

- a 'PIN-NO' as 3.10.1a
- b 'SIG' as 3.10.1b
- c 'DRVR' column indicates if this input connected to an output
- d 'J' not used
- e 'N.P' column indicates if this input is a dummy non-probeable input pin
- f 'SG/DRVR' column
 - i If the input driver is known, the digits 1 and 2 give the driver device number, and digits 3 and 4 give the drivers output pin number.
 - ii If the input is unconnected the 4 digits give the learned signature.
- g 'D/S' as 3.10.1f.

The 'P' command lists out to the terminal or printer all the currently held device data, in the same format as specified above. WIRE-AND devices, which are pseudo devices and are created behind the scenes without the user's knowledge, are listed under the 'P' command. All such WIRE-AND devices have a common device name of "*" which can never be created by the user so there will be no likelihood of confusion.

Any key will stop the 'listing' and, when stopped, any key will restart it. The escape key will return to the command level.

3.11 'S' Strip Connections with Driver/Sensors

This command is activated by the 'S' key and is used to remove all connections in the device data structure between device pins and driver/sensors. Since these connections were made using the FILL command, STRIP can be considered as an 'UNFILL'.

4 GUIDED PROBE DIAGNOSTIC MODE

Once the device data has been learnt and verified correctly the test program can be run with full use of the diagnostic capability. When the program is run and a failure occurs between the "DIAGBEGIN" and "DIAGEND" statements, then the guided probe diagnostic mode is automatically entered.

The user will be asked from the LED display and printer (if 'PRINTALL' on), to probe a series of pins on the board. For each pin request the operator should place the guided probe on the pin and then acknowledge that he is ready by operating the 'PROCEED' key of the test console. The collected signature will be sent to the printer and if the point probed was an input pin then a 'FAULT' or OK message is printed. If the signature is a stuck-at value then 'S @ 0' or 'S @ 1' is also printed.

If the operator slips with the probe while signatures are collected, a 'REPROBE' message will be sent to the display and the probe should be firmly retained at the intended point and the 'PROCEED' key operated again.

After a sequence of probes, and when all faulty nodes have been discovered, the diagnostic algorithm will decide the source of the fault. This will be printed out regardless of whether the 'PRINTALL' switch is active. It will not always be possible to diagnose accurately to a single output node, and in these cases (usually caused by feedback) a series of nodes will be displayed, one of which will be the trouble source. If a failing node is faulty due to being stuck-at-one level, this will also be indicated.

Example probing sequence: TP 010101
 2010 TEST FAILED
 PIN REQD MEAS
 8/A1 LO HI
 DIAGNOSTIC V1.6

```

PROBE U10 PIN 6
B583
PROBE U10 PIN 4
4ACF
FAULT
PROBE U6 PIN 12
4ACF
PROBE U6 PIN 14
REPROBE
F933
FAULT
PROBE U11 PIN 10
F933
PROBE U11 PIN 8
S @ 0
0000
FAULT
PROBE U12 PIN 3
S @ 0
0000
PROBE U12 PIN 1
CABC
OK
PROBE U12 PIN 2
6EFB
OK
PROBABLE FAULT(S)
U12 PIN 3 S @ 0

```

4.1 Operator Misprombes

It is very likely that the operator will make a mistake at some stage in the probing sequence and probe the wrong pin. The diagnostic algorithm tries to detect such misprobes as soon as they have occurred, and prompts the operator to probe the correct point.

If a misprobe is not detected as soon as it occurs and the information is used by the diagnostic algorithm, later detection of the misprobe can cause diagnosis to be abandoned.

Misprobes are detected in several ways, which Figure 5 illustrates.

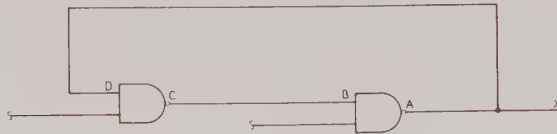


FIGURE 5 - EXAMPLE FOR OPERATOR MISPROBE

Assume that a path from a fault comes along the signal line X.

Pin A is the first place measured and its signature checked to be the same as the faulty signal being followed.

Pin B is then measured and discovered to have a faulty signature, say 1234.

Pin C is measured and should have a signature of 1234. If it does not have this signature, there are several possibilities:

- 1 The track between B and C is broken and the two points are not connected.
- 2 The operator misprobed at B.
- 3 The operator misprobed at C.

The operator is warned that a misprobe has probably occurred and is invited to retry. If he does retry, he is asked to probe B again. This should detect misprobes at B.

If the signature at B is still faulty the operator is invited to probe C again. This should detect misprobes at C.

If the signatures at B and C are still different, the retry will be attempted until the operator gives up, accepting the fact that the track between B and C is broken.

Now that the fault has got back to pin C, pin D is probed. This should have the same signature as that already seen at A.

If it does not, there are several possibilities:

- 1 The track between D and A is broken.
- 2 The operator misprobed at D.
- 3 (Unlikely). The operator misprobed at A and at least one other point connected to A in a way which looked consistent.

The operator is warned of a likely misprobe and asked to probe both points A and D, in that order. Hopefully this will move the probe to different ICs and so cause the correct points to be probed

because the operator is being particularly careful.

The two signatures are used together with the originally measured signature at A to choose between the likely causes of different signatures.

If this unlikely misprobe sequence is diagnosed and the operator does not want to try again, the complete diagnostic procedure is abandoned and must be retried.

Example misprobe with recovery and misprobe/continuity fault detection:

PROBE U6 PIN 7 (recovery)

EDAE

FAULT

PROBE U8 PIN 3

DDD4

MISPROBE-RETRY?

Y

PROBE U6 PIN 7

EDAE

FAULT

PROBE U8 PIN 3

EDAE

PROBE U8 PIN 1

272A

FAULT

PROBE U7 PIN 6

272A

PROBE U7 PIN 4

7DF6

FAULT

PROBE U2 PIN 3

7DF6

PROBE U2 PIN 12

S at 0

0000

OK

PROBE U7 PIN 5

6EFB

OK

PROBE U8 PIN 2

```

CA84
OK
PROBE U6 PIN 5
C082
OK
PROBE U6 PIN 10
1EC8
FAULT
PROBE U8 PIN 6
1EC8
PROBE U8 PIN 4
F6BB
OK
PROBE U8 PIN 5
E873
FAULT
PROBE U12 PIN 8
E873
PROBE U12 PIN 9
350D
FAULT
PROBE U9 PIN 4
7161
MISPROBE-RETRY?
N
PROBABLE FAULT(S)-
CONTINUITY FAILURE!
U12 PIN 9
U9 PIN 4

```

4.2 Feedback and Diagnostic Resolution

If there is no feedback in the circuit, a fault will be detected exactly. When there is feedback in the circuit, any single stuck-at fault which does not propagate right round the loop as a series of stuck signals, will also be detected as the probable fault. However, to cover the few exceptional cases where this is not so, a second 'league' of possible faults is also listed such that if it is found that the probable diagnosis was incorrect then the fault will certainly be in the list of possibles.

Other types of fault in feedback structures may cause diagnosis to the whole of the loop.

Where multiple faults exist in feedback-free circuits, they will be detected one at a time (except in certain very unlikely cases).

Where multiple faults exist in circuits with feedback, no predictions can be made, but if there are stuck-at faults which do not stick complete paths round the loop, one of them at least should be reported as the probable fault.

These predictions ignore the extremely unlikely cases where a faulty sequence occurs which has the same signature as the correct sequence at a node.

Eliminating feedback has two major effects:

- a guaranteeing correct diagnosis and,
- b giving a much shorter probing sequence.

If the board does contain feedback, the following outlines techniques for breaking the loop to improve diagnosis.

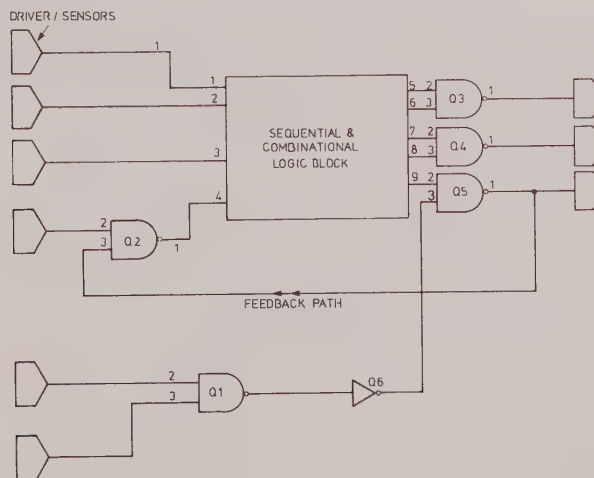


FIGURE 6 - EXAMPLE CIRCUIT WITH FEEDBACK

Consider the circuit shown in Figure 6 which contains feedback around the path from output of Q5 pin 1 to input of Q2 pin 3. The circuit block marked "sequential and combinational logic" represents a "great mass" of random logic gates whose outputs 5 through to 9 are functions of all its inputs 1 through to 4. Clearly any fault in the logic block could propagate around the loop and back into the input giving a potentially very large loop of faulty signatures.

By holding Q5 pin 3 permanently low during the period of the test program from 'DIAGBEGIN' to 'DIAGEND' the output of Q5 pin 1 is always high and the inputs to the logic block are not affected. Q5 pin 3 may be physically clipped low during this time and a separate part of the test program outside the 'DIAGBEGIN-DIAGEND' loop used to 'scope' loop or conditionally pin-state test the remaining untested gates.

Another approach is to hold Q5 pin 3 low by programmed inputs holding Q1 pin 2 or pin 3 low throughout the 'DIAGBEGIN-DIAGEND' loop and then running a second test program where Q5 pin 3 is always held high and testing the remaining logic in the second test program 'DIAGBEGIN-DIAGEND' loop.

4.3 Bidirectional/Bus Structured Devices

Bidirectional/bus structured devices present extra complexity to diagnosis due to the inherent feedback created by the bus pins bidirectionality: see Figure 7. Devices such as microprocessors and D.M.A. controllers, by their nature of driving the address bus as well as being connected to the data bus, can create total feedback structures, which radiate faults to every bus-connected device on the board and further. Such devices which drive the address bus should be removed from the board before test or at the least disabled by a control pin to enable the tester to drive the address bus in their place.

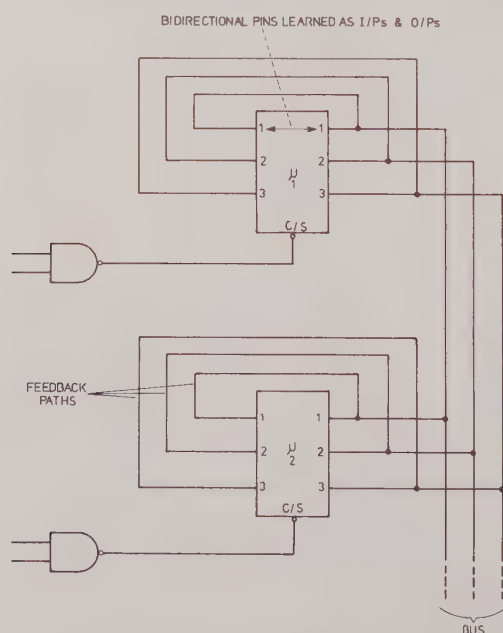


FIGURE 7 - FEEDBACK INHERENT ON BUS STRUCTURED DEVICES

Where there are many devices connected to the data bus, multiple test programs may be used to remove the inherent feedback of the data bus. This should be done by nominating a single device as the bus master such that its bidirectional pins are only learnt as outputs. All other devices are treated as bus slaves and their bidirectional pins learnt only as inputs. A separate test program should then be written for each bus-connected device learnt as bus master.

A problem peculiar to bus structure designs is precise identification of a device that is placing erroneous data onto a bus. If a device is transmitting to the bus when it should not, and the chip selects have not been discovered faulty, it becomes difficult to resolve the fault beyond the bus node. One possible method is to use a current sensing probe to determine on which point current is being sourced or sunk when it should not.

When a fault is found on the bus, unnecessary bus pin probing can be eliminated by use of the dummy input pin attribute to force probing of all chip select generator outputs before any other bus pins.

- a If the bus is being tested as bidirectional, the first device learned on the bus (U2) should have learned all the chip select generator and read/write pins connected to dummy inputs. These should be defined before any real input pins of the bus-connected device.
- b When the bus is being tested unidirectionally, the device acting as bus master (U2 again say) should learn the chip select generator outputs connected to the dummy inputs in the same sequence as for 'a'.

Produced by Walker Industrial Publicity

Printed in England AP/1.84/500

MP MASTER



MA24504